5-2008

Detail Enhancing Denoising of Digitized 3D Models from a Mobile Scanning System

Bradley I. Grinstead
University of Tennessee - Knoxville

To the Graduate Council:

I am submitting herewith a dissertation written by Bradley I. Grinstead entitled "Detail Enhancing Denoising of Digitized 3D Models from a Mobile Scanning System." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Electrical Engineering.

Mongi A. Abidi, Major Professor

We have read this dissertation and recommend its acceptance:

Jian Huang, Andreas Koschan, David Page, Hairong Qi

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a dissertation written by Bradley Ian Grinstead entitled "Detail Enhancing Denoising of Digitized 3D Models from a Mobile Scanning System." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Electrical Engineering.

_____

Mongi A. Abidi, Major Professor

We have read this dissertation
And recommend its acceptance:

_____

Jian Huang

_____

Andreas Koschan

_____

David Page

_____

Hairong Qi

Accepted for the Council:

_____

Carolyn R. Hodges, Vice Provost and
Dean of the Graduate School

(Original signatures are on file with official student records)

Detail Enhancing Denoising of Digitized 3D Models from a Mobile Scanning System

A dissertation
Presented for the
Doctor of Philosophy Degree
The University of Tennessee, Knoxville

Bradley Ian Grinstead
May 2008

Acknowledgements

When you arrive at the end of a season, it is often difficult to remember all those you passed along the way. At the end of this journey, I owe my thanks to many people, most of whom will have to go unmentioned for sake of brevity. For that, I hope they will forgive me. First and foremost, I must dedicate this dissertation to my wife Sabrina, who stood with me throughout the entire process, sparing no effort to help me continue and succeed, even taking on the lion's share of the financial responsibilities for our family to allow me to study unfettered. To my parents goes another large portion of gratitude, for their forbearance of a long educational process and the delays in familyhood the extended education forced upon them. I know they understand, and that alone is enough.

I owe Dr. Mongi Abidi, Dr. Andreas Koschan, and Dr. Andrei Gribok a debt of gratitude that cannot be expressed for their generosity, insight, and help. Without their support, I cannot believe I would have made it to this point. Special thanks indeed go to Dr. Koschan, whose regular technical comments and interesting conversations provided me with much food for thought during these last few years. I credit Dr. Gribok with the origination of the majority of my technical achievements. His knowledge of esoteric methods of analysis set my feet upon the path I eventually would finish. I would also like to thank Dr. Hairong Qi, Dr. David Page, and Dr. Jian Huang for agreeing to serve on my committee and for their time, support, and fruitful commentary. I would also like to thank Matt Heric and the folks at IAVO Research and Scientific for their funding and personal support in the last stage of my Ph.D. efforts.

Finally, I would like to remember the students and staff of the IRIS lab. Faysal Boughorbel, who helped me through my early years and got me playing goalie for the Chinese Army soccer games. Yohan Fougerolle and Michael Roy, who helped me to broaden my knowledge about our European friends as well as providing wonderful technical conversations. Special recognition goes to Sreenivas Rangan Sukumar for true friendship and being a human blackboard to bounce ideas off of on a daily basis. Dr. David Page for his interesting views on the world and technology, as well for his advice on how to approach academic publishing. The administrative staff, Justin Acuff, Vicki Courtney-Smith, Kim Cate, and Diane Strutz for their help and support. Thank you all.

Abstract

The acquisition process of digitizing a large-scale environment produces an enormous amount of raw geometry data. This data is corrupted by system noise, which leads to 3D surfaces that are not smooth and details that are distorted. Any scanning system has noise associate with the scanning hardware, both digital quantization errors and measurement inaccuracies, but a mobile scanning system has additional system noise introduced by the pose estimation of the hardware during data acquisition. The combined system noise generates data that is not handled well by existing noise reduction and smoothing techniques

This research is focused on enhancing the 3D models acquired by mobile scanning systems used to digitize large-scale environments. These digitization systems combine a variety of sensors – including laser range scanners, video cameras, and pose estimation hardware – on a mobile platform for the quick acquisition of 3D models of real world environments. The data acquired by such systems are extremely noisy, often with significant details being on the same order of magnitude as the system noise. By utilizing a unique 3D signal analysis tool, a denoising algorithm was developed that identifies regions of detail and enhances their geometry, while removing the effects of noise on the overall model.

The developed algorithm can be useful for a variety of digitized 3D models, not just those involving mobile scanning systems. The challenges faced in this study were the automatic processing needs of the enhancement algorithm, and the need to fill a hole in the area of 3D model analysis in order to reduce the effect of system noise on the 3D models. In this context, our main contributions are the automation and integration of a data enhancement method not well known to the computer vision community, and the development of a novel 3D signal decomposition and analysis tool. The new technologies featured in this document are intuitive extensions of existing methods to new dimensionality and applications. The totality of the research has been applied towards detail enhancing denoising of scanned data from a mobile range scanning system, and results from both synthetic and real models are presented.

Table of Contents

List of Figures

List of Tables

List of Publications

Accepted for Publication

B. Grinstead, A. Koschan, and M. Abidi, "Developing detailed *a priori* 3D models of large environments to aid in robotic navigation tasks," Proceedings of the *SPIE: Unmanned Ground Vehicle Technology VI*, vol. 5422, pp. 561-568, 2004.

B. Grinstead, A. Koschan, and M. Abidi, "Fast digitization of large-scale hazardous facilities," Proceedings of the *International Conference on Robotics & Remote Systems for Hazardous Environments*, pp. 521-525, 2004.

B. Grinstead, A. Koschan, and M. Abidi, "Hybrid self localization for a mobile robotic platform in indoor and outdoor environments," *Transactions of the American Nuclear Society*, **92**, pp. 52-53, 2005.

B. Grinstead, A. Koschan, and M. Abidi, "A comparison of pose estimation techniques: Hardware vs. video," Proceedings of the *SPIE: Unmanned Ground Vehicle Technology VII*, vol. 5804, pp. 166-173, 2005.

B. Grinstead, A. Koschan, D. Page, A. Gribok, and M. A. Abidi, "Vehicle-borne scanning for detailed 3D terrain model generation," *SAE 2005 Transactions Journal of Commercial Vehicles*, pp. 196-204, 2005.

B. Grinstead, A. Koschan, A. Gribok, and M. A. Abidi, "Outlier rejection by oriented tracks to aid pose estimation from video," *Pattern Recognition Letters*, **27**(01), pp. 37-48, 2006.

B. Grinstead, A. Koschan, A. Gribok, and M. A. Abidi, "Improving video-based robot self-localization through outlier removal," Proceedings of the *ANS Joint Emergency and Response/Robotic Systems Topical Meeting*, pp. 322-328, 2006.

B. Grinstead, A. Koschan, D. Page, and M. A. Abidi, "Model building for simulation and testing under uncertain conditions," Proceedings of the *SPIE: Modeling and Simulation for Military Applications*, vol. 6228, pp. 98-109, 2006.

B. Grinstead, A. Koschan, and M. A. Abidi, "Geometry refinement of 3D surfaces using kriging," Proceedings of the *3D Data Processing, Visualization, and Transmission*, 2006.

B. Grinstead, S. Sukumar, D. Page, A. Koschan, D. Gorsich, and M. A. Abidi, "Mobile Scanning System for the Fast Digitization of Existing Roadways and Structures," *Sensor Review*, **26**(04), pp. 283-289, 2006.

Submitted and Pending Review

B. Grinstead, A. Koschan, D. Page, and M. A. Abidi, "Automated spatial refinement for irregularly sampled 3D surfaces," *under review IEEE Transactions on Visualization and Computer Graphics*, 2006.

1 Introduction

1.1 Motivation

With the recent advances in robotics and computer graphics technologies, 3D models of large areas are becoming increasingly valuable information for a variety of applications: Robotic couriers need “maps” of their routes; robotic security systems can be enhanced with an as-built layout of the facilities they patrol; driving simulators can be improved with realistic models of the environments being simulated; virtual tours of museums and city downtown districts can be developed from as-built structures.

Traditional methods for populating these types of virtual environments involved teams of graphics artists constructing the scenes from a library of pre-defined objects combined with freehand manipulations, or from a rough measurement skeleton. More recently, photogrammetry has been used to acquire these models *in situ* using cameras and classical stereo vision theory, but with varied results and at the expense of a large amount of time. However, technology has now advanced to the point where we can use lasers in conjunction with color cameras to quickly and accurately acquire the desired 3D models. In effect, we can now *digitize reality*.

What do we mean by *digitizing reality*? We use this term to describe the process of acquiring geometrically correct of photorealistic 3D digital models of real world objects or scenes, and processing the data so that it can be presented in a meaningful fashion, according to the needs of the specific application. Within the framework of this proposal, we are focused on the digitization of large-scale environments. In essence, digitizing these environments entails:

- A system of hardware for acquiring the 3D geometry.
- A system of high-resolution cameras for acquiring the texture overlays used for the models.
- Software algorithms for processing the acquired data, including functions such as: data fusion for multiple sensors, data analysis, noise removal, model simplification, etc.

The scanning of large environments poses a number of constraints on the design of the acquisition system. These constraints are often application-specific – what resolution is necessary? – but a few apply to any task. One of the main concerns for the system is to be able to acquire the necessary data with minimal impact on the target environment. For example, how many city governments would want a 3D model of the downtown area if they knew it would take 4 weeks of having the streets closed to all traffic? Not many, to be sure. This is why we have developed our system to quickly digitize large-scale environments, using a mobile scanning platform.

The acquisition process of digitizing a large-scale environment produces an enormous amount of raw geometry data. This data is corrupted by system noise, which leads to 3D surfaces that are not smooth and details that are distorted. Any scanning system has noise associate with the scanning hardware, both digital quantization errors and measurement inaccuracies, but a mobile scanning system has additional system noise introduced by the pose estimation of the hardware during data acquisition. The combined system noise generates data that is not handled well by existing noise reduction and smoothing techniques.

Figure 1.1a shows a typical 3D surface generated by a mobile scanning system. Here, the noise present on the surface is due not only to the noise of the range scanner, but also to system noise introduced by the pose estimation of the hardware during data acquisition. Simple smoothing techniques such as Laplacian smoothing can reduce the "bumpiness" (Figure 1.1b) of the surface, but tend to oversmooth the data, leaving the corners rounded and the details obscured. More involved filtering techniques such as anisotropic smoothing can remove some of the noise effects, while preserving the detailed areas (Figure 1.1c), but even these fail to produce good results when the details are on the same order of magnitude as the system noise.

Ideally, we would like to have a denoising process that will identify the regions of interest and enhance the details present in those areas, while removing the noise effects from the surface. Such a system might produce a result like that shown in Figure 1.1d, where the logo "Jared" can be clearly seen, while the lettering underneath "Galleria of Jewelry" is also evident. The research presented here is focused on developing just such a detail-enhancing denoising procedure for 3D surface models generated by our in-house mobile scanning system. It is important to note at this stage that for this research we are interested in metrology (accurate measurement on geometry) rather than visualization (graphics). There are many techniques in the Computer Graphics fields that can be followed to "build" surface appearance from photographs/video without an accurate geometry foundation, but these do not apply in this case.

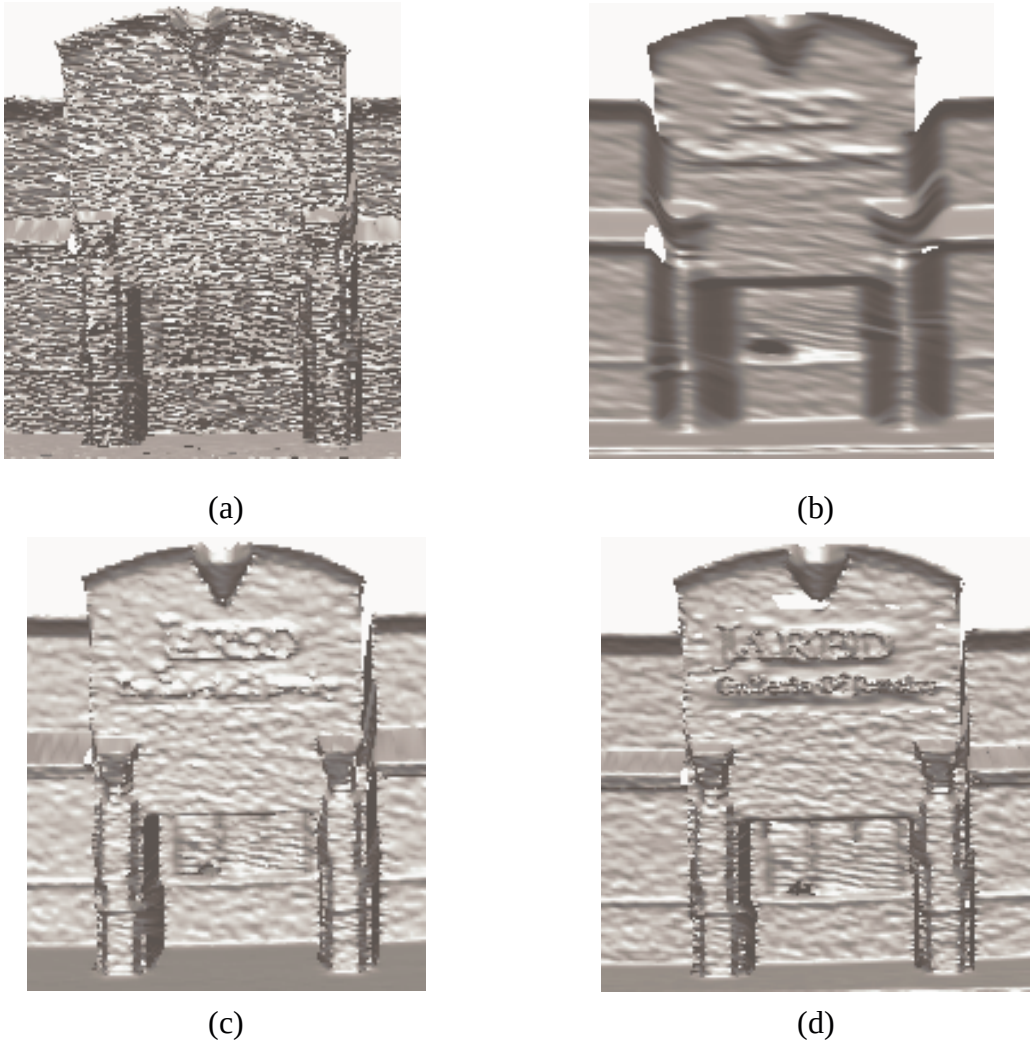


Figure 1.1. Motivation for detail-enhancing denoising. (a) Original mobile scanned data of the Jared's building. (b) Simple Laplacian smoothing applied to the original data. (c) Anisotropic smoothing applied to original data. (d) Ideal results from a detail-enhancing denoising of the Jared's model.

1.2 Areas of Contribution

Our research presents an auxiliary contribution in the area of pose estimation from video and several primary contributions in the area of 3D model post-processing in the form of detail-enhancing denoising, under the framework of a large-scale mobile scanning system (as shown in Figure 1.2):

- **A New False Match Detection Algorithm to aid Pose Estimation From Video**

  We designed an algorithm for detection and removal of false matches in the Pose from Video (PfV) process that is based on the observation of the behavior of the correct feature matches for scenes undergoing the motion common to our mobile scanning platform. The motion of our scanning system causes correct feature tracks to run parallel to each other. Our algorithm identifies those feature tracks that share a common behavior, and filters out the false matches that do not conform.

- **Automated Detail Enhancement through Modified Kriging**

  In order to enhance the details of a region in an acquired dataset, additional sampling needs to be applied to the region of interest. Re-acquiring the data at a higher resolution may not be an option for many applications, so we propose to add additional data in the region of interest through interpolation. Kriging is a data interpolation technique that is ideal for the anisotropic data common to mobile scanning systems. For this research, parameters for a modified form of kriging are automatically identified, and the surface in the area of interest is resampled to provide a higher level of resolution in those regions.

- **A Extension of Empirical Mode Decomposition to 3D Surfaces**

  One of the main goals of this research is to extend the state-of-the-art in the 3D surface analysis field. There is currently a large hole in the state-of-the-art in the area of signal decomposition and analysis for 3D data. We intend to fill that gap with an extension of a powerful 1D signal decomposition and analysis tool – Empirical Mode Decomposition (EMD) – to signals attached to 3D surfaces. In so doing, our research in this area will provide a framework for identifying surface frequencies for the denoising process.

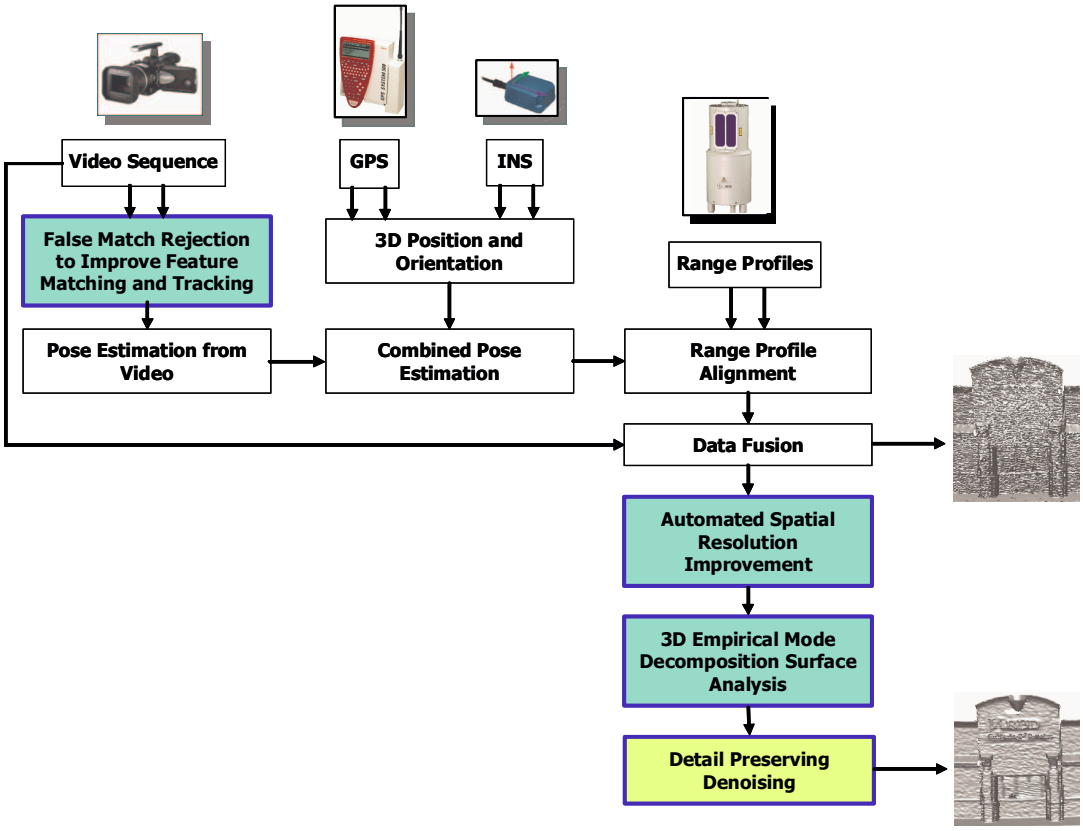


Figure 1.2. General flowchart for our mobile scanning system, including both the data acquisition and the post-processing. Blue colored boxes indicate areas of contribution in our research. Yellow indicates application of the research to real world problems.

The ultimate application of this research is to combine the individual contributions into a framework for detail-enhancing denoising. The input for this process will be a noisy 3D surface, as scanned by a laser scanner, and the output will be a denoised version of the original data. This denoised version will have the advantages of a smoothed model, with the noise impact removed or greatly reduced, while the regions of high detail (logos, hard edges, etc.) will be enhanced and protected from the smoothing process.

1.3 Document Layout

The remainder of this document is arranged as follows:

- **Chapter 2** presents some related work done on topics covered by this thesis work, including: mobile range scanning, pose estimation systems, data regularization, surface interpolation, and 3D signal decomposition and analysis.

- **Chapter 3** discusses our mobile range scanning system and it's component parts: range scanners, pose estimation, and model creation.

- **Chapter 4** presents an overview of pose estimation from video, with a focus on the Oriented Tracks algorithm we have developed for improving the true/false match ratio of the feature matching process.

- **Chapter 5** covers the use of automated Kriging for detail enhancement of 3D models.

- C**hapter 6** discusses our proposed extension of Empirical Mode Decomposition to 3D surfaces.

- **Chapter 7** describes the application of 3DEMD to the signal source separation problem.

- **Chapter 8** describes the application of 3DEMD to provide a detail enhancing denoising solution for noisy, large-scale 3D data.

- **Chapter 9** concludes with a discussion of tasks proposed for this dissertation.

2 Related Work

Depending on the application and the environment to be digitized, many images from various types of sensors may be needed to correctly model the target scene. Thus, digitization of reality begins with the acquisition of data from various sensors. This collection of sensors can include vision sensors – such as range, video, or thermal cameras – as well as localization and pose estimation sensors – such as GPS, INS, and gyroscope systems. The next step is to register the data from the various sensors into a common coordinate frame for processing. Data fusion is then necessary to combine the multi-modal data into a cohesive whole, which is ready for visualization or further processing. The ideal digitization system would be one that could enter a target area and quickly, accurately, and automatically generate a digital 3D representation of that environment.

2.1 Digitizing Large Scale Models

There have been many systems developed in the last few years with the goal of obtaining accurate urban models from real world environments. These systems can be divided into two basic categories: (1) the image-based approach, where 3D digitization is performed via 2D imagery, and (2) the range-based approach, where 3D geometry is directly acquired from laser range scanners. Figure 2.1 shows these two categories, and how we divide them into mobile and stationary systems.

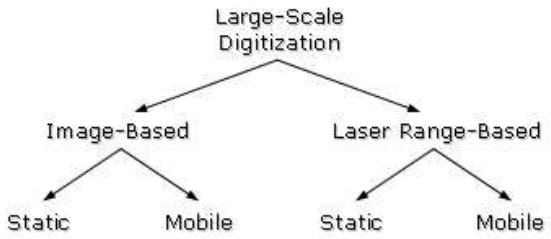


Figure 2.1. Two categories of large-scale digitization systems, with their mobile and static sub-groups.

2.1.1 Image-based Approaches

Image-based approaches utilize stereo- or motion-based techniques to infer geometrical structure and are also known as indirect approaches. Examples of 3D digitization based on 2D imagery include the MIT City Scanning Project (Antone & Teller, 2000), where a system has been developed to automatically reconstruct visible geometry using spherical mosaic images. Each camera's position and orientation (pose) is first estimated using positioning sensors (Teller et al., 2001), and then refined through an image matching process. Feature correspondence and edge analysis are used to extract the geometry of the scene. Each node is entered into an adjacency map to provide global reference information. The full 3D model of the area is created by linking the 3D geometry from each node with its neighbors', according to the adjacency information. Additional high-resolution imagery can be used for further texturing (Coorg & Teller, 1999) and vegetation modeling (Shylakhter et al., 2001).

Gruen and Wang developed CyberCity Modeler to generate the planar surfaces of building from imagery, and combine them with digital terrain maps to provide a 3D model of an entire city (Gruen & Wang, 1999). The building models were generated from manually picked interest points and Photogrammetric tools. Bauer, et al. developed Metropogis (Bauer et al., 2002), a feature based city modeling package. Their approach was to extract edges from the image series, determine the vanishing points, and use the vanishing point set to reconstruct the surfaces in 3D. This approach allowed them to quickly reconstruct a series of planar surfaces suitable to modeling most urban scenes.

Airborne platforms were some of the first to utilize Photogrammetric methods on mobile platforms to digitize large-scale models. Aerial scanning systems based around the assumption that roof features were planar were some of the first systems. Geometry was inferred by grouping coplanar lines and corner features and using a bundle adjustment (Bignone et al., 1996; Frere et al., 1998; Moons et al., 1998). Baillard and Zisserman (Baillard & Zisserman, 1999) extended this by a planar-sweep partitioning method to refine the models and fit them more closely to the imagery. Suveg and Vosselman (Suveg & Vosselman, 2002) used a hypothesize and fit approach to improve the results on very complex buildings. Enhancement of the appearance and fit of these extracted models has been investigated, including: adding texture (Wang & Hanson, 1997; Gruen & Wang, 1999; Jaynes, 1999; Lee et al., 2002), identifying windows and doors (Wang & Hanson, 1997), and identifying vegetation (Brenner & Haala, 1998; Straub & Heipke, 2001).

Ground-based mobile systems have also been used to develop 3D models from video sequences of urban environments. Zisserman et al. developed a "VHS to VRML" system that used handheld commercial camcorders to acquire 3D models of urban environments. Features between pairs of images were matched using an "F-tracker" guided by the estimated geometry (Zisserman et al., 1999). Lines are then matched across image triplets and the results refined through bundle adjustment. The resulting projection matrices are then used to calculate the 3D geometry. Uehara and Zen (2000) used similar structure-from-motion techniques to extract the geometry from a ground-level video system mounted on a vehicle. The image streams are referenced using GPS data to provide positioning information.

Pollefeys, et al. (2000) retrieved a 3D scene model from a sequence of uncalibrated video images. Using an auto-calibration process, the authors follow a sparse feature matching process, followed by a geometry projection and refinement, a dense feature matching stage, and finally 3D reconstruction (see Figure 2.2).

Some groups have focused exclusively on the development of mobile imaging systems for urban scanning. Epipolar Plane Images (EPIs) (Gotoh et al., 1999; Notomi et al., 2000) were investigated to utilize the regular structure of urban features to define the relationship between image pairs, and the resulting transforms used to compute the 3D geometry of the scene. Faugeras et al. (Faugeras et al., 1998) used the regular features in urban environments to compute 3D geometry from an uncalibrated system, using the point matching and geometry reconstruction approaches described above. Camera calibration is done through identifying the plane at infinity, deriving the affine structure, and then calculating the Euclidian projection.

Other efforts involved using a mobile robot – or combination of robots – and stereo vision techniques to build a traversal map of an environment, including rudimentary geometry. One such effort used a collection of small robots with single onboard cameras to cooperatively navigate and map a facility (Thayer et al., 2001). Relative pose estimation between the robots is done by combining the internal instrumented pose information with a robust feature matching procedure based on invariant moments.

These approaches show that it is possible to use indirect methods to digitize real-world geometry. However, the difficulties inherent in stereo matching, and the limited resolution of digital color cameras, make it difficult to reconstruct realistic 3D models with the desired level of accuracy and resolution.

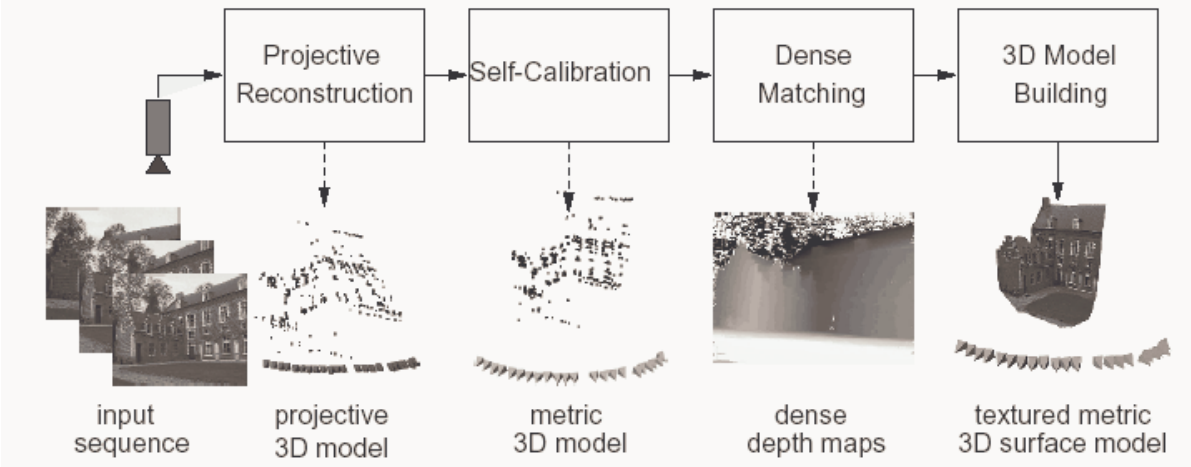
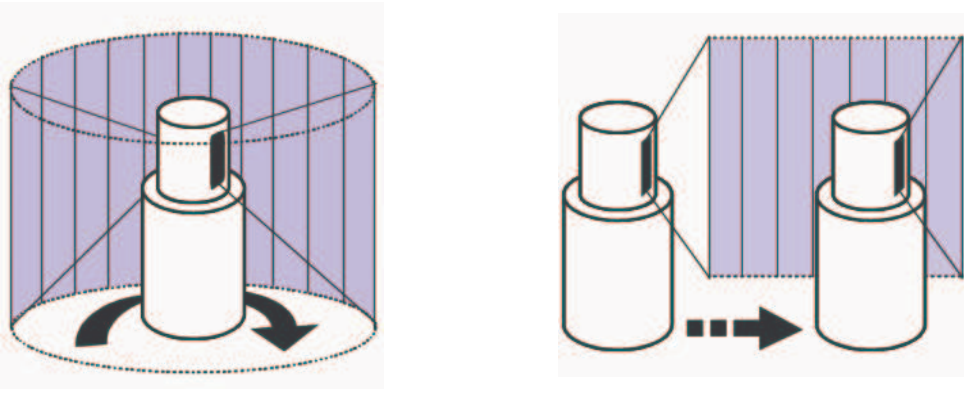


Figure 2.2. Automatic calibration-based system from 3D from Video as described in (Pollefeys et al., 2000).

2.1.2 Range-based Approaches

The second category of methods acquires 3D geometry of real-world scenes in the more direct manner. Specifically, this category consists of methods involving the acquisition of 3D geometry information directly, via range scanners. Range-based modeling of 3D environments is itself divided into two categories: stationary (radial) scanning, and mobile scanning. In radial scanning – the more traditional range scanning approach – the range scanner is in a fixed location, and the scanning head rotates about a given axis to generate the collection of profiles known as a range image, as seen in Figure 2.3a. In the mobile scanning approach, the scanner is mounted on a mobile platform and the range image is acquired by moving the platform past the scene to be digitized. Figure 2.3b demonstrates the mobile scanning concept.

Stamos and Allen (2000) developed a radial scanning system for outdoor model acquisition. Their system is ground-based and utilizes the Cyrax scanning system to acquire the range data. The data is segmented into planar patches, and 3D lines are extracted and used in registering the range scans. Volumetric sweeps are then used to fill out and merge the registered scans. El-Hakim, et al. (1997) and Sequeira, et al. (1999) have developed radial range scanning systems to acquire indoor geometries in great detail. In the former case, the scanning platform moves to a new position, acquires visual and laser range data, and registers them using a bundle adjustment approach for pre-calibrated systems. Sequiera, et al. used a similar approach, with line features present in both the visual images and the range scans being used for sensor registration and data fusion. The stop-scan-move approach was also used by Surmann, et al. when they developed their indoor exploration robot. The robot uses a laser range scanner to acquire the geometry of the area around it. Then, a next-best-view



(a) (b)

Figure 2.3. Two modes of acquiring geometry with a laser range scanner. (a) Radial scanning has a “cylindrical” field of view, while (b) Mobile (vehicle-borne) scanning has a “planar” field of view.

algorithm identifies the next location for scanning, and a path planning algorithm works out the best way to get to that point. The multiple scans are registered using a standard ICP approach (Besl & McKay, 1992). Zhao and Shibasaki (2001c) utilize a long-range radial laser range scanner to digitize urban scenes. To register the individual range scans, the authors developed the Z-image by projecting all range points onto the ground plane. Simple pattern matching is then used to align individual scans into a common framework.

Jokinen (1997) developed an airborne system for outdoor scanning that used a hierarchical low-to-high resolution registration technique. Simultaneous registration of all views was used to minimize the cumulative errors inherent in sequential techniques. Piecewise planar surfaces are built for each view and the views are merged in the overlap areas. In this case, the range views were taken from an aerial platform. Brenner and Haala (1998) also utilized an airborne laser range scanner to develop 3D models of urban areas. They use a digital cadastral map to match "ground truth" footprints of buildings to their scanned geometry to identify buildings and correct the geometry. In addition, they categorize several types of vegetation from both the range and the visual data they collect. Vosselman and Dijkman (2001) use prior knowledge of the structures, combined with plane fitting techniques to enhance the models from their airborne system, while Maas (Maas, 1999) uses a combination of morphological filters, histogram analysis, and weighted moments to identify and enhance building shapes, roof gables, etc.

Zhao and Shibasaki (2001b) performed urban modeling by utilizing two orthogonally mounted range scanners on a ground-level vehicle scanning system that incorporates GPS/INS/odometer readings into its sensing package. This data is used to provide an initial coarse alignment of range profiles, which is then refined using scan-line matching on the range profiles from the horizontal scanner. Früh and Zakhor (2001a) have developed a similar system using two orthogonal range sensors. Bearing and velocity sensors are used to provide a coarse registration between profiles, with refinement by scan-line matching. In addition, the vehicle path estimation is enhanced by Monte Carlo localization from aerial maps of the scanning area. These mobile systems utilize an orthogonal scanline matching technique for pose estimation, and thus restrict themselves to a planar urban environment.

Georgiev and Allen (2004) combined INS, GPS, and video to provide the location and orientation of a mobile robot in an urban environment. They provided an *a priori* map of local structures to be matched to by the laser range scanner and pose from video systems, with the ability to update the structure maps from the scans acquired by the laser range scanner.

2.2 Self-localization

For our purposes, self-localization refers to the process of identifying the motion of the sensor package relative to some basis. In this work, we also interchangeably use the term pose estimation to denote this process. When used in a robotics sense, this process is often referred to as determining the ego-motion of the platform. We use the terms interchangeably.

In particular, we are attempting to estimate the position and orientation (attitude) of the sensor package on our mobile platform.

There has been much research effort put into determining the ego-motion of a mobile platform over the last 30 years or so. Instrumented approaches are quite common, where global positioning systems (GPS), inertial navigation systems (INS), wheel encoders, and other devices physically measure the position and orientation of the sensor package. This is known as "direct" pose estimation, as the position and orientation parameters are specifically measured. Indirect methods of pose estimation include the scanline matching methods and pose from video methods.

2.2.1 Instrumented Self-localization

Typically, instrumented approaches for localization rely on GPS data for position information, and INS data for determining the orientation (Bretz, 2000). Fusion of the two datasets can be done via Kalman filtering. Kalman filtering takes advantage of the similar characteristics of INS and GPS data to provide an integrated ego-motion estimate, with performance superior to that of either individual system. Kalman filtering tracks the drifting parameters of the system to provide accurate estimates of the system's position, orientation, and velocity.

Using instrumented approaches to define the scanner's pose has been done for many years. Airborne scanning applications use GPS/INS packages exclusively to estimate the pose of the cameras during the scanning process (Bossler & Schmidley, 1997; Frere, et al., 1998; Moons, et al., 1998). Further processing can then be done to correct for the inter-scanline registration errors (Fricker et al., 1999; Crombaghs et al., 2000). Nygårds, et al. have implemented a fusion of GPS, INS, and digital compass information for their Unmanned Aerial Vehicle (UAV) project (Nygårds et al., 2004).

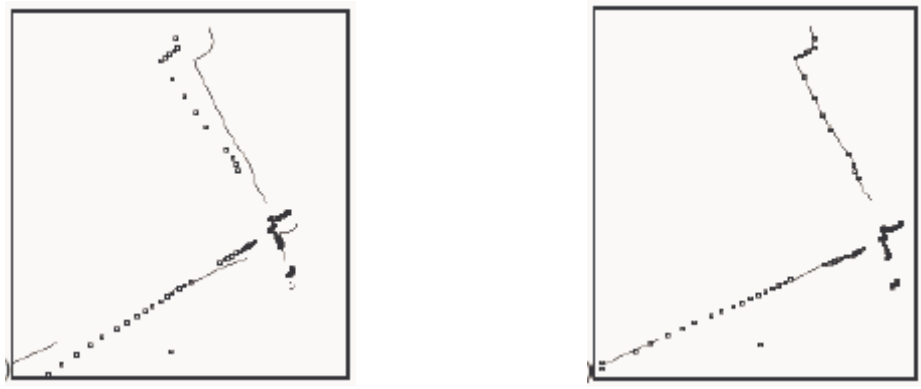
Instrumented approaches have also been used on ground-based vehicles to determine the ego-motion of robotic systems. Cui and Ge (2003) use an Extended Kalman Filter (EKF) combined with linear path approximation to estimate the ego-motion of a vehicle in a cluttered urban environment. Julier and Durrant-Whyte (2003) have integrated a vehicle model into their EKF-based pose estimator, which showed to have improved performance in areas where GPS signal quality was low. Masson, et al. (2003) integrated GPS, INS, bearing and laser range data using an EKF to provide accurate positioning information, even in the cases where one or more of the sensors are giving unreliable data. Direct pose estimation has also been used in ground-level urban scanning systems such as those of Zhao, Shibasaki, and Manandhar (Zhao & Shibasaki, 2001a; Manandhar & Shibasaki, 2002).

2.2.2 Self Localization And Mapping

In 1991, Leonard and Durrant-White introduced the concept of Self Localization and Mapping (SLAM), where both the map of the area and the vehicle's position in it must be estimated (Leonard & Durrant-Whyte, 1991). This is often done in robotic systems, where repeated horizontal range scans of an environment are matched together to provide the motion the sensing platform underwent between scans. At the same time, the system builds a comprehensive "map" of its environment to aid in matching further scans. This process of matching successive scans to the online model can be seen in Figure 2.4. The current map is shown in light gray, while the newest scan is shown as black dots. Matching the newest scanline to the model involves finding the transformation that brings the new data into agreement with the model. The rotation and translation represented by this transformation is the motion undergone by the scanning package.

In the last decade, many extensions have been made of the original SLAM theory to different environments and purposes. Simon, et al. developed an extension of the SLAM algorithm to a single line range scanner (Simon et al., 1994), where the individual scans were matched to the developing model via the Iterative Closest Point algorithm (Besl & McKay). The urban scanning projects of Zhao and Früh (Früh & Zakhor, 2001b; 2001a; Zhao & Shibasaki, 2001b; Früh & Zakhor, 2002) use orthogonally mounted laser range scanners and the SLAM methodology to simultaneously capture urban geometry and perform pose estimation. Autonomous systems have even been developed to map both indoor and outdoor environments, using robotic platforms as small as a 1' x 1' x 1' cube (Hähnel et al., 2003).

Unfortunately, due to the fact that this scanline matching process is restricted to the plane of the laser scanner, at best the process can only provide positioning within a single plane, and



(a) (b)

Figure 2.4. SLAM localization process. The translation and rotation needed to align the two scans is the motion that the vehicle underwent between them. Taken from (Früh & Zakhor, 2001b).

orientation only about the plane's normal, restricting its use to a planar environment such as hallways indoors, or a relatively flat outdoors environment such as a parking lot. Früh and Zakhor (2002) have used their vertical laser scanner and the assumption that the buildings being scanned should be orthogonal to the plane of motion to attempt to remove this planar environment constraint to some degree of success. The laser-based SLAM formulation has even been extended to any form of mobile mapping and self-localization, such as using full 3D corridor scans to perform the SLAM duties (Surmann et al., 2003). Hähnel, et al. use two different configurations for their laser SLAM approach (Hähnel, et al., 2003). The first is the orthogonal system similar to those above, and the second is a single scanner mounted on a pan-tilt unit. Histogram analysis and planar approximations help their system efficiently store and process the 3D map.

2.2.3 Video-Based Self-localization

In order to eliminate the need for expensive equipment to measure the pose of the sensor package directly, and to overcome the limitations of SLAM, many researchers have chosen to estimate the platform's pose via a video sequence, known as pose from video (PfV). This is done by identifying certain features between successive images in a video sequence, matching them between the images, and using stereo vision techniques to identify the transformation necessary to align the images to common reference frame. This transformation can then be used to obtain the ego-motion of the platform.

There have been several surveys on methods for determining 3D motion from a 2D image sequence (Jebara et al., 1999; Salvi et al., 2001; Armangue et al., 2003), which categorize the PfV methods into those that infer the motion of the camera from the Epipolar Constraint, and those that are based directly on the optical flow. They evaluate the performance of several of these motion estimators within the framework of a mobile robot moving in a planar environment.

Branca, et al. (Branca et al., 1998) use Moravec's operator to find features in an underwater environment. These features are then matched using a radiometric similarity metric. The 2D motion field determined by these matching feature tracks is projected into 3D space, and the motion parameters are determined by minimizing

$$E = \left\| V(x,y) - \sum_{i=1}^{6} (c_i \psi_i(x,y)) \right\| \tag{2.1}$$

for the 3D motion parameters: $c_1 = T_x$, $c_2 = T_y$, $c_3 = T_z$, $c_4 = R_x$, $c_5 = R_y$, $c_6 = R_z$, where the ψ_is are the 3D projection basis functions (Branca et al., 2000).

Cho, et al. (Cho et al., 2001) also use an optical flow technique for their ego-motion estimation scheme. An iterative non-linear approach is used to solve for the 3D motion parameters, using the 2D motion vectors as input. A weak perspective projection is used along with a guess of the average depth of the scene as an initial guess to the iterative solver.

Results on both real and synthetic data show their method to outperform the Total Least Squares method of solving the overdetermined system.

Morency and Gupta (Morency & Gupta, 2003) also use an optical flow technique, based on defining the 2D feature tracks, projecting those features into 3D space, and using an ICP-type approach for minimizing the distances between the projected points. A normal flow constraint is added to minimize the differences based on both appearance and depth information. Another use of normal flow developed the camera's motion parameters in a probabilistic framework (Roy & Govindu, 2000).

Faugeras (Faugeras, et al., 1998) presented a standard Structure from Motion (SfM) technique based on epipolar constraints. Corner features are detected and matched via an intensity correlation. Projection matrices are found by identifying the epipoles of the images, and a Euclidian reconstruction is performed with *a priori* knowledge of the camera's intrinsic parameters. Pollefeys, et al (Pollefeys et al., 1999 2000, Pollefeys, 2001) extended this method to uncalibrated cameras by automatically calibrating the camera online.

Ego-motion estimation for a space lander has been accomplished using epipolar based techniques (Johnson et al., 2000). Interest points are identified and tracked using the STK feature tracker (Shi & Tomasi, 1994) and the projections between image pairs are found using the method discussed in (Hartley & Zisserman, 2000). The uncertain scale factor in the motion estimate is eliminated by use of the onboard laser altimeter, allowing the full 6DoF motion estimate to be made.

Vidal, et al. (Vidal et al., 2001) have developed a batch method of determining the pose of a camera undergoing motion, using epipolar constraints. From a batch of images in a sequence, normalized epipolar constraints are formalized, and a geometric optimization is used to solve for the motion parameters of the system. The batch normalized epipolar constraints are formulated using LaGrange multipliers, and evidence given by the authors shows that this formulation can outperform standard bi-focal epipolar constrained motion estimation.

Another batch processing ego-motion model uses a sliding window of three images and trinocular epipolar constraints to define the camera motion (Zhang & Shan, 2003). Interest points are extracted, using the Harris operator, and intensity correlation is performed to match the triplet features. The accuracy of trinocular stereo is enhanced via bundle adjustment, determining the projection that minimizes the distance between the points from all three images in 3D space. Zisserman et al. (Zisserman, et al., 1999) also use image triplets for epipolar geometry estimation. In this case, point features are used to guide the initial transformation, with refinement being performed using line features.

Martins, et al. (Martins et al., 2003) proposed a method of determining 3D orientation of a moving camera in an urban environment without feature matching or 2D motion detection. Instead, they use the *Manhattan World* assumption – i.e., in an urban environment, most objects have edges that correspond to the 'global' x-, y-, and z-axes – and detection of orthogonal lines in the image to determine camera orientation. A similar approach was used

by Liu, et al. (Liu et al., 2003), who detected 3D corners in an image sequence by identifying orthogonal line intersections. The 3D corners are then tracked through the image sequence, and the 3D motion of the camera is inferred directly.

Recently, there have been several attempts to combine instrumented pose estimation with PfV, mostly in the area of autonomous robotics. Corke (2004) uses a standard epipolar constraint method to estimate the pose of a small, autonomous helicopter, and then combines that estimate with inertial data using a straightforward low-pass filter. Alenya et al. track contours in an image sequence, and combine the PfV information with inertial updates using an EKF. They report that real-time control of an autonomous vehicle is practical using their system, with verifying experiments using a mobile cart.

Graovac (2004) utilizes a landmark point tracking procedure, combined with inertial estimates, to develop a hybrid pose estimation approach. Characteristic point tracking – initialized by human interaction – is used to define the video navigation routine. Focus of Expansion (FOE) and Center of Rotation (COR) are identified with respect to these landmark points. The inertial navigation system is added to improve the pose estimates when the landmark points are within the field of view, and is used as a standalone pose estimator during those times when the landmark points are out of the field of view of the camera.

Lobo (Lobo et al., 2003) used inertial data to define the FOE and COR, which is then passed to the video navigation system as an initial guess for the motion estimator. A vertical reference for the local gravity vector is also determined from the inertial data and, in combination with ground plane detection, makes the image-to-image registration process a 2D problem. The corresponding depth maps are transformed into this vertical frame of reference and the ego-motion parameters estimated.

2.3 Surface Interpolation

In general situations, often the small-scale details that are most affected by standard denoising algorithms are the ones we most wish to preserve. In some cases, not only do we wish to preserve these data, but enhance them as well, bringing out the information they contain. In essence, the enhancement is done by locally adding in more information – extrapolated from the measured data – increasing the resolution in the area of interest.

Interpolation is the process of improving the resolution of data by placing estimated measurements in between known measurements. This is done by using the measured data at known locations to estimate the values at the new, unknown locations. Interpolation techniques that give estimates at the known locations equal to the measured value are known as ‘exact interpolators’. All other interpolation methods are ‘inexact interpolators’, otherwise known as ‘approximators’ because the values estimated at the known locations are approximately equal to the measured values. The quality of data fit of these approximators is usually measured with a statistical metric, such as Sum of Squared Distances (SSD).

The majority of interpolators are local methods, where the estimated value is assumed to have a close relationship to those measured points nearby, and less relationship to points further away. The interpolation process involves:

- Defining the appropriate region of influence around the point to be estimated;
- Finding the measured points that lie in this region;
- Choosing a model to represent the functional surface that best fits this set of measured points; and
- Using that model to estimate the value for the point of interest

There are a number of common interpolation methods that are used to refine data, including (but not limited to): nearest-neighbor, linear, moving average, inverse distance weighting (IDW), and radial basis functions (RBFs). We will briefly cover these techniques, as well as their drawbacks for the data specific to our system, with an alternative to these techniques discussed in the rest of the section.

Nearest neighbor interpolation assigns the value of the closest measured point to the point of interest. This interpolation is used more for categorical values than numerical ones, and almost never for estimating surface values. Linear interpolation fits a linear function (i.e., a plane) to the three points nearest the point of interest. Along the surface of the triangle formed by the three known points, the surface function is assumed to have a linear dependence, and two linear approximation equations are all that is needed to evaluate the point of interest. This approximation leads to a surface that is piecewise continuous, but not smooth. A smoother version is seen in bilinear approximation, where the 4 closest points to the location of interest form a quadrilateral on the projection plane. A bilinear surface is fit to these four points, and then the value of the unknown point can be calculated from the parametric surface.

Inverse distance weighting estimates the value of the unknown location by weighting the impact of nearby points strongly, while lessening the influence of data points farther away. It is perhaps the simplest form of neighborhood interpolation. The estimate at the unknown point is evaluated as

$$\hat{f}(x) = \frac{\sum_i d_i^{-p} \bullet f(x_i)}{\sum_i d_i^{-p}}, \tag{2.2}$$

where x_i is the known point in the neighborhood of the unknown point x, and d_i^{-p} is the is the p-th power inverse Euclidian distance from x to x_i. It should noted that in practice the size of the neighborhood is much smaller than the number of measured data points.

The power p defines how much difference in effect the measured points have throughout the entire neighborhood. If p is too large, the function falls off rapidly with distance, tending

towards the nearest neighbor approximation. If p is too small, there is not much variance in the effect of locality on the weighting, tending towards the average neighborhood value.

Radial basis functions follow an approach similar to IDW in that the known points in the neighborhood of the unknown exert an influence that depends on the radial distance from the unknown location. However, instead of a simple power of the Euclidian distance, RBFs use distance-dependant functions that are more complex in an attempt to better fit the surface local to the point of interest. RBF interpolation estimates the unknown value as a sum of weighted radial basis functions

$$f(x) = P(x) + \sum_i w_i \varphi(d_i), \tag{2.3}$$

where $P(x)$ is the bias of the surface at the unknown location, w_i are the weights applied to each RBF, $\varphi()$ is the user-specified RBF, and d_i is the Euclidian distance between the ith neighbor and the point of interest (Wahba, 1990). In addition, a regularization parameter is usually added to the system to provide robustness to measurement noise. RBF interpolators work well with smooth data, but tend to have difficulty with sharp, small-scale features (Carr et al., 2001; Dinh et al., 2002).

Subdivision surfaces define a smooth surface as the limit of a sequence of successive refined triangular (polyhedral) meshes. The refinement is done by dividing each triangle into 4 smaller triangles, with the new vertices determined according to the subdivision scheme. Loop subdivision is an approximation scheme based on the generalization of quartic triangular splines (Loop, 1994). The division step is followed by a smoothing step, where a Gaussian smoothing operator moves the new vertices into position. This sequence is guided by weights chosen to ensure continuity of surface curvature or tangents.

Dyn, et al. proposed an interpolating subdivision scheme called Butterfly subdivision for a regular mesh (Dyn et al., 1990), which has been generalized for meshes of arbitrary topology by Zorin, et al. (1996). In this case, the expressions for the weights of a vertex change depending on the neighborhood of the vertex in question. By design, the various subdivision schemes generate smooth surfaces (Taubin, 1995b), and occasionally some of the small-scale details present in the scanned data are obscured. Features such as boundaries and sharp edges can be preserved by marking them as fixed and modifying the associated weights (Zorin, et al., 1996).

Alternatively, Karbacher, et al. use the assumption that the underlying surface of a mesh is represented by a set of circular arcs (Karbacher & Häusler, 1998; Karbacher et al., 2001), which can be locally refined to preserve sharp features. This method assumes that a good estimation of the surface normals is available, even if the data points themselves are noisy. Figure 2.5 shows how the circular arcs method can be used to subdivide a surface by projecting the midpoints of each triangle edge onto the arc associated with that edge.

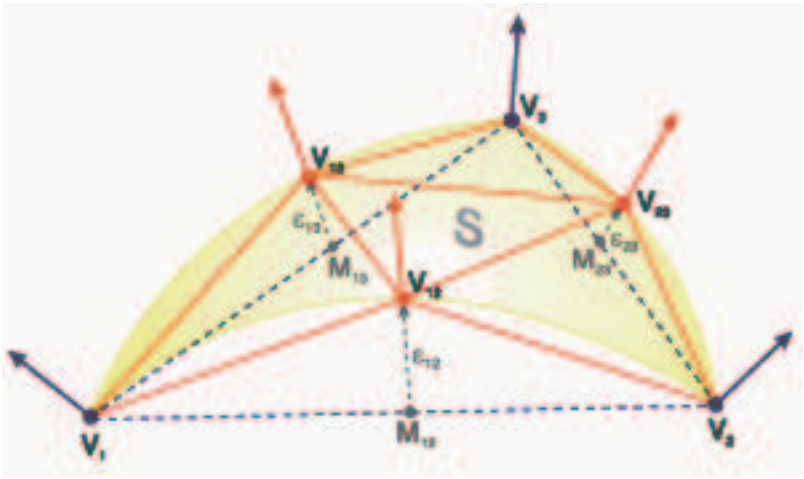


Figure 2.5. Subdivision of a triangle using the circular arcs approach, courtesy of (Karbacher, et al., 2001).

None of the above methods provide direct measures for the evaluation of their "goodness of fit" to the data. The only way to evaluate the goodness of the prediction is to compute estimates for a set of known data points that were not used in the original interpolation. Another problem with these methods is that there are a number of parameters which must be chosen, and there is no way of knowing *a priori*:

- The size, shape, or orientation of the neighborhood used to perform the refinement
- The number of measured points required to give the best representation of the data
- The appropriate weighting scheme for the neighborhood points, and which distance-based function will give the best results

There is another option for surface interpolation that is not restricted by the above questions. Kriging is a geostatistical method for refinement of spatial data (Cressie, 1991) that makes use of regional variable theory to define a stochastic process for data interpolation. Observed data drives the process, which is optimized to reduce the variance of the estimation errors. Thus, kriging provides an interpolation that: makes no assumptions about the form of the data being estimated, sampling criteria, density, shape, etc.; has a built-in estimate of the uncertainty of the estimate; and is ideal for non-uniform, sparsely sampled data. More detail on the kriging process is given in Chapter 5.

Because the kriging is based on the measured codependency of data points, it can provide a more accurate local representation of the data than polynomial- or function-based methods. Figure 2.6 shows the interpolatory refinement process on a simple signal using a variety of interpolation techniques. A signal with a step-discontinuity is sampled in a non-uniform fashion. For the refinement process, new data points are estimated at the midpoints between samples. Linear interpolation yields a simple "connect the dots" results, with no

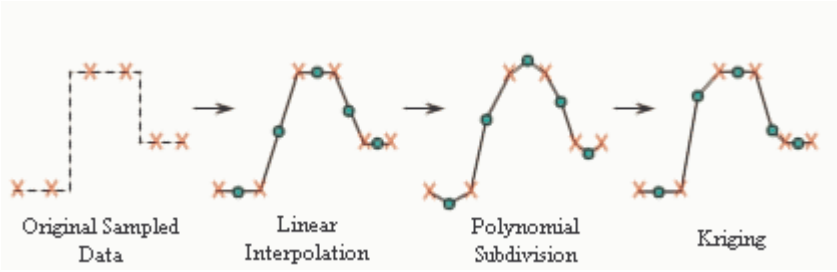


Figure 2.6. Interpolatory refinement of a step-discontinuity.

enhancement of the data. Polynomial subdivision fits a polynomial (typically cubic) to the measured points and moves the estimated values to minimize the specified energy functional. This yields a signal that is closer to the original step, but is rounded due to the smooth energy functional that is the driving component of the subdivision. Kriging uses the local co-dependencies to generate a closer representation of the original signal, albeit with some smoothing introduced. This behavior of kriging is what leads us to use it as the basis for the local surface refinement and detail-enhancing algorithm discussed in Chapter 5.

2.3.1 Data Quality

"Quality" as it applies to data has many definitions, but the most common relates to "fitness for use" (Chrisman, 1983), which relates the quality of the data to the use to which it is intended in both current and future settings. Data quality is strongly related to the application use and connot be independently measured (Chrisman, 1991; Strong et al., 1997). Data quality is multidimensional, involving estimates of: accuracy, precision, error, uncertainty, and validation. No matter the effort taken to reduce the errors introduced in the data collection and management processes, errors will always exist in data acquired from real experiments (Maletic & Marcus, 2000). Thus data validation and improvement remain an important stage in the experimental results process.

The improvement of data quality involves two primary stages: error prevention and error correction. In many cases, error prevention is seen as the superior, due to the fact that no error detection and correction method can ever be 100% accurate (Dalcin, 2004). However, in many cases, pre-existing datasets only allow for error correction as a post-processing stage, as reacquiring the data would not be cost effective or even possible.

This is the foundation of the major benefit of using Kriging as the basis for our surface interpolation method. Kriging has a built-in "goodness of fit" criteria, which allows us to see immediately how well the new surface fits the underlying data, and to make a human evaluation of the significance of how good that fit is for the application at hand.

2.3.2 Texturing Methods

The field of Computer Graphics is rife with techniques geared to trick the eye into seeing shape and texture where there is none on the geometry. Run any current game and you will find surfaces that appear rough, textured, complex, curved, or shiny, while in actuality they are low(er) resolution models with graphics processing applied to fool the eye into believing that structure is present. These methods are based on texturing and are the core of any computer graphics programmer's handbook.

Bump mapping was introduced in 1978 (Blinn, 1978) and is a technique to make surfaces appear wrinkled or bumpy. Bump maps can be used to simulate surfaces that would take many polygons to model geometrically. The premise behind bump mapping is that the appearance of wrinkles on a smooth surface can be achieved by modifying how light interacts with the surface - through its normals. Figure 2.7 shows and example of this process by applying a texture to a smooth sphere, with the result appearing textured like an orange. Bump mapping can be combined with light mapping, textures, and shaders to provide intricate appearing textured surfaces.

For bump mapping, the height field texture is used to alter how the light field interacts with the surface, but does not alter the geometry itself. Displacement mapping (Cook, 1984) is a process by which a height (or displacement) field is generated and the underlying surface is physically displaced according to the texture. Displacement mapping requires a dense underlying tesselation of the surface, but the requirements are no more stringent than those supplied by a range scan of an object. Perhaps the biggest obstacle to displacement mapping are the functions required to generate the displacement map.

For our purposes, texturing methods do not provide the requirements necessary for a successful mobile scanning system. The reason is that for many applications, accurate geometry is required, not just the appearance of accurate geometry. For example, in robotic navigation the robot needs to know where exactly a handle is to open the door, including the handle's dimensions and offset from the geometry around it. This is why we chose to utilize range scanners in the first place: for their ability to densely and accurately measure the geometry of a surface. Of the computer graphics techniques, displacement mapping has the most applicability to the task at hand, but the difficulty in deriving the displacement maps makes it only suitable as an alternative form of geometry enhancement for large-scale datasets.

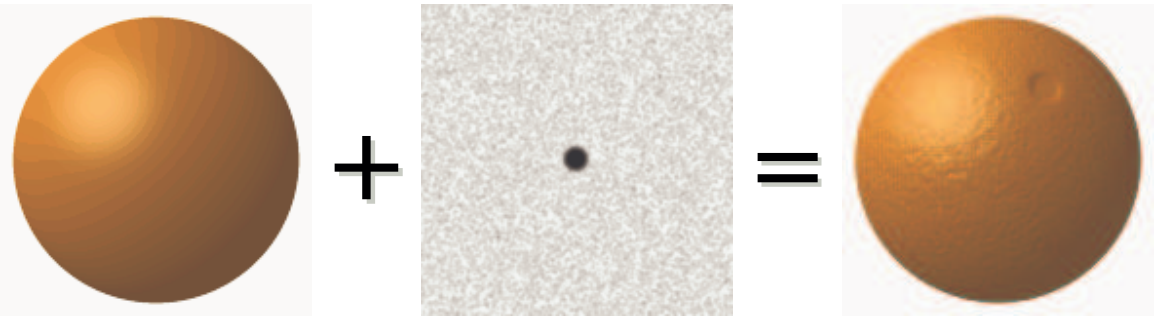


Figure 2.7. Example of the bump mapping process. Images courtesy of wikipedia's entry on bump mapping (http://en.wikipedia.org/wiki/Bump_mapping).

2.4 3D Surface Decomposition and Analysis

Data analysis and interpretation is an important step towards designing processing methods to improve constructed models. In the 3D case, this means analyzing a 3D surface in preparation for further processing, such as: denoising, smoothing, enhancement, filtering, etc. Typical solutions for signal analysis in the 1D and 2D cases – Fourier, PCA, Wavelets – have no direct analogue to 3D surfaces, so other decomposition and analysis methods are needed.

Decomposition of the signal into sub-signals is the essence of most data analysis algorithms, so we will focus our efforts on decomposition methods for 3D surfaces in this thesis. The basis for all decomposition algorithms is to break the signal down into a set of details and an approximation (Mallat, 1999; Roy et al., 2003). The decomposition is then recursively applied to the approximation until the decomposition is halted. To reconstruct the signal, the details are combined with the approximation at each level. Because the approximations each represent the surface at a different level of resolution, these signal decomposition methods are often referred to as multiresolution analysis methods.

Work has been done on developing decomposition techniques in the framework of 3D surfaces over the last 10 years or so. Lounsbery et al (1997) developed the Subdivision Wavelet Transform that utilizes the Loop subdivision framework and a wavelet derivative to decompose a semi-regular 3D mesh with subdivision connectivity. Further work has been done by Zorin et al. (Zorin, et al., 1996; Peng et al.) that utilizes the Loop and Butterfly subdivision schemes to develop mesh editing tools, such as deformations, surface refinement, and smoothing. Peng et al. utilized Loop Subdivision as the marker used for decomposition (Peng, et al., 2001). The authors add another dimension to the multiresolution pyramid scheme by introducing the use of directional filters to capture local edge information. This technique only works well on semiregular meshes, but has yielded some good results in terms of decomposition. These schemes use polynomial approaches for fairness of fit of the subdivision refinement. Kobbelt (1996) developed a variational subdivision scheme where

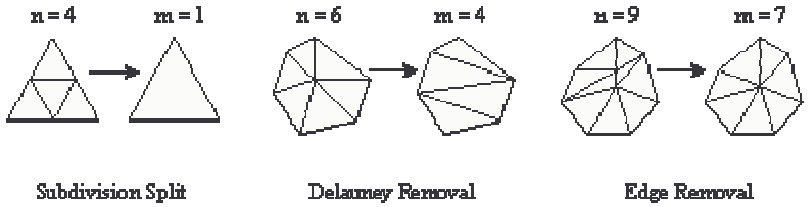


Figure 2.8. Illustration of local triangle decimation schemes.

the driving functions have some physical meaning, such as strain energy, and which improves the fairness of the generated curves. Later, the Variational Subdivision algorithm was extended to handle non-uniform surfaces (Kobbelt & Schröder, 1998).

The previous approaches all make use of traditional subdivision surfaces – each parent triangle split into 4 smaller triangles – for their multiresolution analysis. Other researchers have defined multiresolution analysis schemes for more general 3D surfaces. Bonneau used a Haar wavelet-like transformation to compute the details between a given surface and it's approximation by Delaunay triangulation. Guskov et al. (1999) developed a multiresolution framework for 3D surface decomposition based on the Progressive Mesh framework and a Quadric Error Metric for determining the local details at each level of the decomposition. Within this framework, the authors have shown that they can provide smoothing of 3D models based on the details within the hierarchy of decomposition, similar to analogous decomposition-based smoothing in 1D and 2D. Roy, et al. (Roy, et al., 2003) extended this method to work on attributes as well as on irregularly sampled triangular meshes. They used their extension to provide smoothing of meshes, as well as other applications such as: bandwidth based filtering, denoising using estimates of the system accuracy, and adaptive simplification.

With these decomposition methods, vertices are removed at each level of the decomposition as shown in Figure 2.8. Thus, there is no knowledge of how that vertex would affect the signal approximations at the lower levels of the decomposition hierarchy. The additional knowledge missing in these decompositions provide enough information for more intelligent post-processing tools (Huang et al., 1998). Chapter 6 discusses our proposed 3D decomposition method that is being developed to fill the gaps in the 3D surface decomposition theory.

2.5 Challenges to the State of the Art

A Mobile Scanning System by its very nature has system noise that degrades the model even further than whatever sensor noise may be present. Figure 2.9 shows this effect on one of our data sets – the BILO supermarket. The building entrance has been scanned in both the

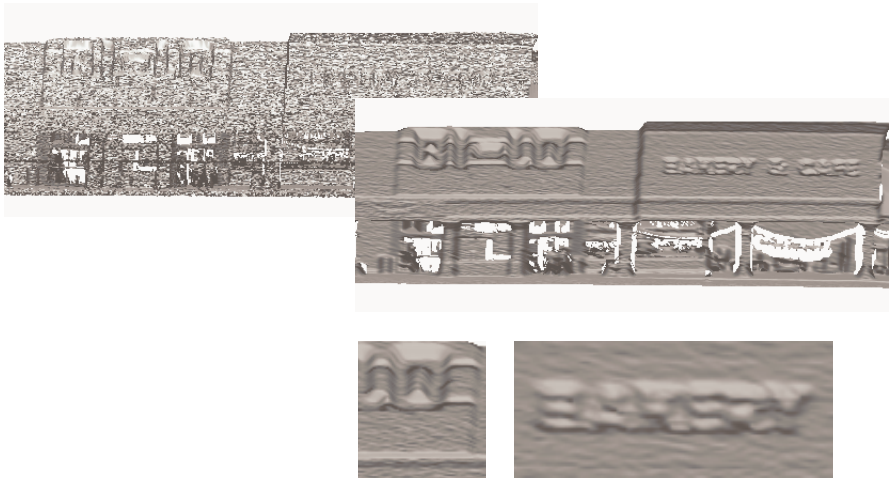
traditional fashion – radial scanning – and by our Mobile Scanning System. Figure 2.9a shows the building surface as scanned by a radial scanning system – both the raw data and the Laplacian-smoothed version. The same sensor was used on our Mobile Scanning System to generate the surface seen in Figure 2.9b. The difference in quality between the two surfaces is evident in the smoothed versions. The letters on the "BILO" and "Bakery & Café" signs are crisper in the radial scan, and the vertical struts in the "BILO" area are not evident in the mobile scan data.. These differences are typical of the tradeoff between data quality and speed of data acquisition that comes from mobile scanning solutions.

Why is this the case? Let's take a look at the sources of error for each data set. The radial scan has noise from the laser range scanner, which is essentially only along the laser line – the line from the scanner's optical center to the surface – and therefore 1-Dimensional. For the mobile scan, the surface is not only corrupted by the scanner noise, but also by the "unknown" position and orientation of the scanner at each profile and the motion of the vehicle.

The pose estimation procedure discussed in Chapters 3 and 4 has uncertainties associated not only with each sensor (GPS, INS, Video, etc.) but also with the fusion of these sensing modalities. In addition to these pose uncertainties, there is also the fact that the horizontal (inter-scanline) resolution is not fixed for the scanner, but depends on the motion of the vehicle. The faster the scanning platform moves, the wider the inter-scanline distance, and the worse the horizontal resolution. This is what causes the vertical struts to appear as missing in the mobile scan data. The fact is that while the vertical (intra-scanline) resolution is the same as that from the radial scan data, the horizontal resolution is 2-3 times worse.

Thus, the ultimate goal of this research is to alleviate the effect of the uncertainties in the pose estimation as well as those from the undersampled nature of the horizontal resolution. In effect, to remove the effects of the system noise on the data, as well as utilize what data we do have to rebuild the structures that the system may have undersampled. The ultimate result would be to have mobile scanned data with the same quality as radially scanned data, with the benefit of not having to go through the effort of acquiring, registering, and merging many radial scans.

(a)



(b)

Figure 2.9. Comparison of noise effects between radial and mobile range scanning. (a) Raw and Laplacian-smoothed surface of the entrance scanned by a radial scanning system with insets showing interesting geometry. (b) Raw and Laplacian-smoothed surface of the same building, scanned by our Mobile Scanning System with insets showing interesting geometry.

3 Mobile Scanning System

Scanning a large environment – such as a city street, or a large building complex – poses a number of constraints in the design and implementation of a 3D scanning system. Some of the constraints are application dependent (e.g., what resolution is required of the model?), while others apply to any application. Perhaps the main concern is to be able to acquire representative data with minimal impact to the environment being scanned. After all, how many cities would want a 3D model of their downtown areas, at the cost of having to close all city streets in a 3-block radius for two weeks? Also, once we have the digitized models, what do we do with them? Even our GHz processors and gigabytes of memory are not sufficient to be able to store, process, and visualize models of hundreds of millions of triangles. In order to meet the demands of many scenarios, we have designed a scanning system that can drive through urban areas at normal driving speeds, acquire data at cm-level resolution, and process the digitized models for visualization.

Our approach is to acquire cm-level resolution models from laser range scans, with the scanner mounted on a moving platform. This allows us to acquire 3D geometry in a variety of environments with varying lighting conditions. If textured models benefit the application, we also capture high-resolution digital imagery in conjunction with the laser range data. Meanwhile, pose estimation hardware is acquiring positioning and orientation information for the vehicle. This information is used to align the individual profiles and bring them into a common coordinate system. Figure 3.1 shows the mobile scanning system components, consisting of a high-resolution laser range scanner, high-definition digital cameras, and pose estimation hardware.

3.1 Range Scanning

The next few paragraphs are a quick review on laser range scanning, specifically concerning time-of-flight laser range scanners. Modern 3D scanners use a number of acquisition techniques such as stereo vision, structured light, laser triangulation, and time-of-flight (Trucco & Verri, 1998) for sampling target objects. Each application provides constraints that guide the choice of scanning technology. Such constraints include: accuracy, distance to target, sampling resolution, sampling frequency, etc. For large-scale outdoor environments, time-of-flight laser range scanning provides a good mixture of accuracy, sampling resolution, and distance to target.

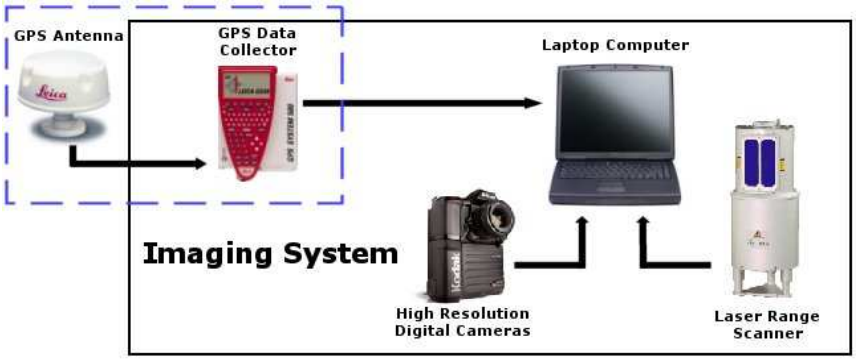


Figure 3.1. Mobile scanning imaging system consisting of a laser range scanner, high-resolution digital cameras, and a pose estimation sensor pack, all of which communicate with a portable computer.

Figure 3.2 shows the process for time-of-flight laser range scanning. A laser diode sends out a coded pulse of infrared light (1), which is deflected by a set of rotating mirrors (2) towards the target object. When the laser beam strikes a surface (3), some of its energy is dispersed while some of it returns to the scanner to be measured by a photosensor (4). Timing electronics (5) keep track of the amount of elapsed time from laser pulse initiation to sensed return. Distance is obtained as a function of the time taken to travel to the surface and back, thus the term *time-of-flight*. The rotating mirrors deflect the laser beam in one axis to obtain a profile of range points. A frame scan can be acquired by rotating the entire system about another axis (Riegl-USA, 2002).

Our scanner is based on time-of-flight laser scanning, supplemented by instrumentation for estimation of the scanner's position and orientation. The scanner used is a Riegl LMS-Z210, with a vertical field of view of 80° and a maximum range of 350 m. The scanner is mounted on the vehicle's roof, such that its scanning plane is perpendicular to the vehicle's direction of travel, as seen in Figure 3.3a. As the vehicle moves past the scene, the scanner acquires 3D geometry in a profile fashion (Figure 3.3b). The sequence of profiles makes up the complete range image for the target scene. The system is modular is design, with the sensors used for geometry scanning, texturing, and even positioning to be changed depending on the target application, with a minimal of changes in the algorithms applied to perform the data fusion and post-processing.

During motion the laser range scanner acquires a profile of the geometry present in the scene, while the pose hardware captures the vehicle's motion. Let [*X Y Z*] be the global coordinate reference for the digitized scene. This can be either a georeferenced coordinate system or a local equivalent. Let [*x y z*] be the laser scanner's internal coordinate system and let the orientation parameters for the scanner be [*roll pitch yaw*]. Each point *P* in the current profile has a 3D identity of $v_P(x, y, z)$, as seen from the scanner, and the scanner has a 3D identity

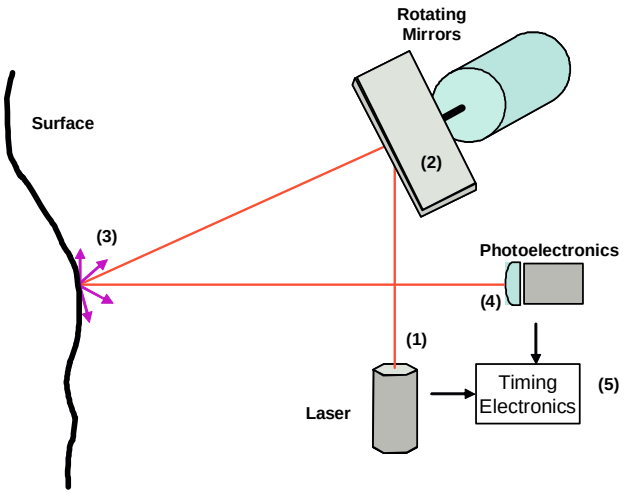
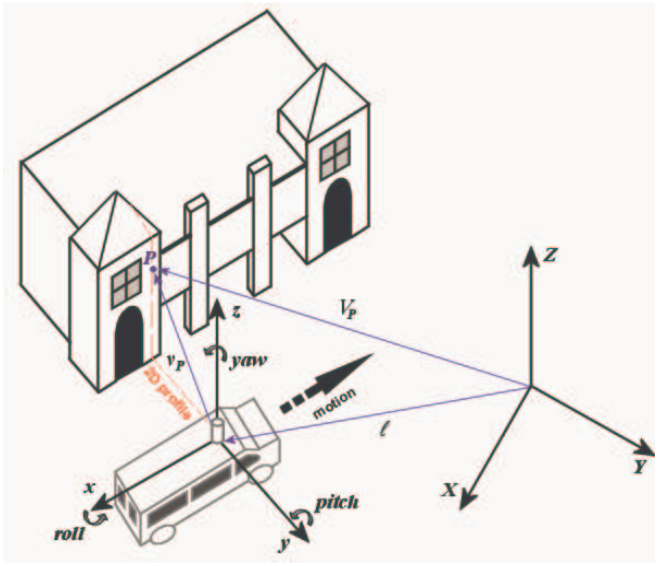


Figure 3.2. Time-of-flight laser range scanning principle.

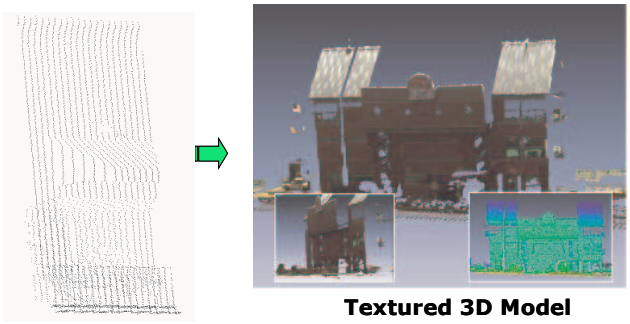
of $\ell(X, Y, Z,)$ in the global reference. Thus, the location of P in the world reference can be found as:

$$V_P = \ell + R \cdot v_p , \tag{3.1}$$

where R is the rotation matrix generated from the *roll*, *pitch*, and *yaw* values determined from the pose estimation system.



(a)



(b)

Figure 3.3. Mobile scanning system. (a) System diagram for the mobile scanning system showing profile acquisition and vehicle motion. (b) A sequence of 2D range profiles captures a building's geometry and, when taken together, make up a 3D model (shown here with color texture).

3.2 Pose Estimation

While the laser range scanner is acquiring geometry, the pose estimation system is capturing all of the positioning and orientation information for the vehicle. This information is used to align the individual profiles and bring them into a common coordinate system. Pose estimation instruments include: Global Positioning Systems (GPSs), Inertial Navigation Systems (INSs), odometers, velocimeters, etc. GPS, though a common positioning solution, has a few drawbacks. A single system can only attain a real-time kinematic accuracy of a few meters and loss of satellite signals due to canyon (building) shadow can cause loss of positioning data. The accuracy can be improved by using two systems in concert to perform differential positioning, while shadow effects can only be overcome with additional information.

In our setup, the scanner's location in the global reference is determined via a Leica differential GPS system that is capable of positional accuracy of up to 2 cm and an update rate of 10 Hz. The orientation parameters for the system are acquired using an Xsens IMU, with an update of 100 Hz and sub-degree accuracy. IMUs utilize gyroscopes and linear accelerometers for acquiring orientation and positioning information, as seen in Figure 3.4. Due to the drift errors inherent in inertial systems, the positional accuracy degrades significantly over long periods of time, but using the GPS position as a keyframe where available, we can provide rough positional information with the IMU even in areas where GPS fails (e.g., urban canyons, under foliage, etc).

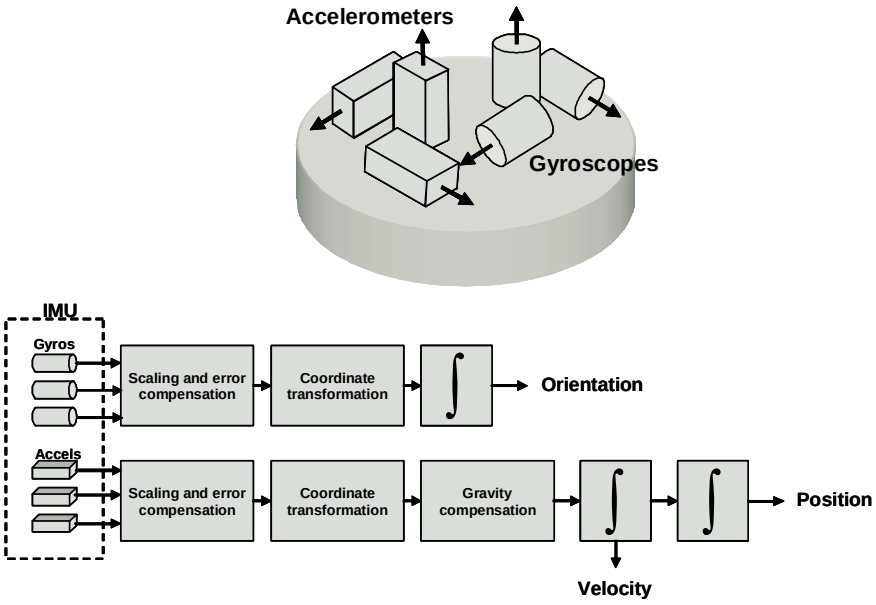


Figure 3.4. Basic inertial system components and navigation algorithm – adapted from (Grewal et al., 2001).

Using instrumentation to measure the position and orientation of the system provides accurate results, within the operating parameters of the instruments themselves, but sometimes the mobile scanning system will need to be used in areas where direct measurement of the platform's pose will not be feasible. Thus, we are investigating indirect methods of pose estimation, a discussion of which can be found in Chapter 4: Pose Estimation from Video.

3.3 Experimental Results

For our experimental system, we used a Riegl LMS-Z210 laser range scanner to acquire the geometry. This scanner has a maximum range of 350 m and an 80° vertical field of view, with a range accuracy of ± 5 cm and an acquisition rate of 10,000 points/sec. The vehicle position was determined via a differential GPS system with an accuracy of 2 cm at 10 points/sec, and the orientation was measured with an XSens MT9 IMU at 100 Hz. For each of the presented results, the vehicle was driven past the scene at normal driving speeds.

Figure 3.5 shows the results of acquisition for a small strip shopping center. The complete model was acquired in 2 minutes and consists of over 3 million triangles. Color imagery was combined with the geometry to provide a photorealistic model. Figure 3.5a shows a sample digital image of the scene that was digitized (Figure 3.5b). Figure 3.5c is a closer view of the digitized model.

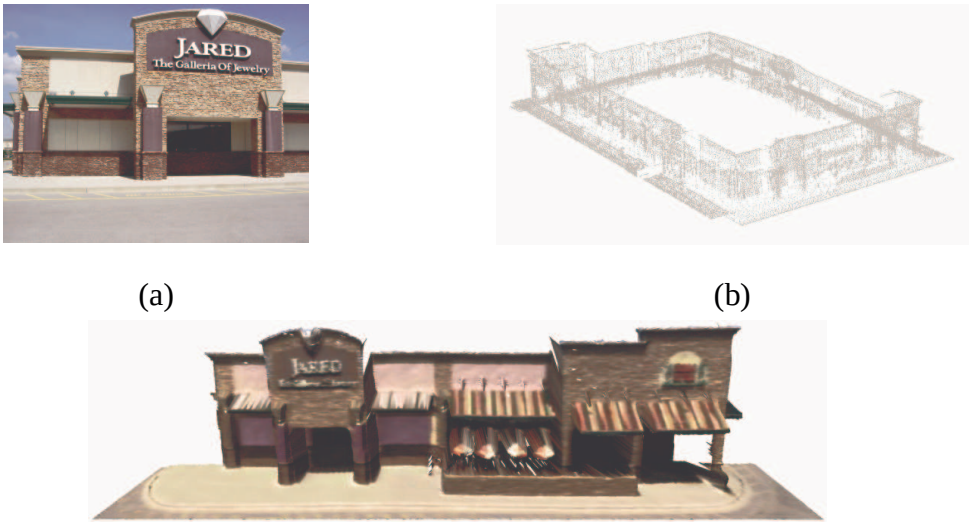
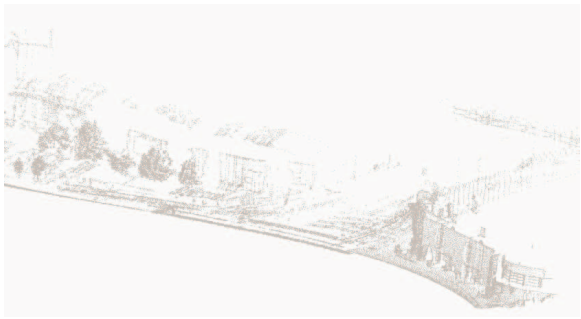


Figure 3.5. Strip mall sequence. (a) High-resolution digital image of a portion of the building. (b) Low resolution 3D geometry in point cloud form. (c) High resolution textured 3D model of one side of the complex.

Figure 3.6 demonstrates the use of the system on a larger scale. The data shown here is a 2-block subset of a dataset that encompassed 3 kilometers of scanning. The data shown here was acquired in three minutes and contains over 5 million triangles. The focus of these results is on the Women's Basketball Hall of Fame, a prominent landmark in Knoxville, TN. Figure 3.6b shows the geometry acquired for this subset (as a cloud of points).

To demonstrate the use of the Mobile Scanning System on a city-wide scale, we scanned the West Town Mall in Knoxville, TN. The dataset is rather large, consisting of over 5,000,000 data points, 5,000 color images, and 48,000 pose measurements. However, only 18 minutes of scanning time was required to acquire the data, which encompassed over 1.5 km of scanning. Figure 3.7 shows a few results from this dataset. Figure 3.7a shows an aerial view of the West Town Mall area, with the vehicle's scanning path overlaid. The raw geometry is shown in reduced point cloud form in Figure 3.7b, and the textured 3D model shown in Figure 3.7c is a small subset of the entire dataset.

(a)

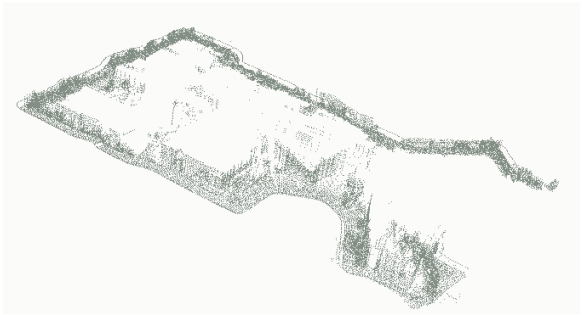


(b)

(c)

Figure 3.6. Women's Basketball Hall of Fame sequence. (a) Aerial view of scanned area. (b) Scanned geometry (shown in low resolution) for the road segment highlighted in blue. (c) 3D model of Hall of Fame complete with texture.

(a)



(b)

(c)

Figure 3.7. 1.5 km scan of the Downtown West Mall in Knoxville, TN. (a) Aerial view of the scanned region, with vehicle path overlaid. (b) Scanned geometry in low resolution point cloud form. (c) Textured subset of the full model.

4 Improved False Match Rejection By Oriented Tracks

4.1 Pose estimation From Video

With the advent of inexpensive video cameras, optical navigation is being increasingly utilized on robotic vehicles (Shapiro et al., 1994; Johnson, et al., 2000; Pollefeys et al., 2002; Armangue, et al., 2003). Using video for navigational purposes involves determining the interframe motion between successive images, and often relies on the matching of feature points from one image to another. Accurate and timely estimation of the robot's motion is dependant on the quality of these feature matches.

It is known that two images of a static scene are related to each other through their epipolar geometry (Barnard & Fischler, 1982; Faugeras, 1993), and epipolar geometry constraints have been used to perform Structure and Pose from Motion tasks (Chua et al., 2000; Hartley & Zisserman, 2000; Forsyth & Ponce, 2003). A robotic vehicle's pose can be determined from a video sequence by finding corresponding features between adjacent images in the video sequence and using the scene's epipolar geometry to calculate the position and orientation changes between the two images (Branca, et al., 2000; Rivlin et al., 2003; Usher et al., 2003; Chroust & Vincze, 2004).

The pose estimation from video process can be seen in Figure 4.1. First, distinctive features in an image pair are identified. Next, the features in the first image are matched to corresponding features in the second image. Given a good set of correspondences, the motion state of the camera can be calculated, up to scale, using a two stage motion estimation algorithm.

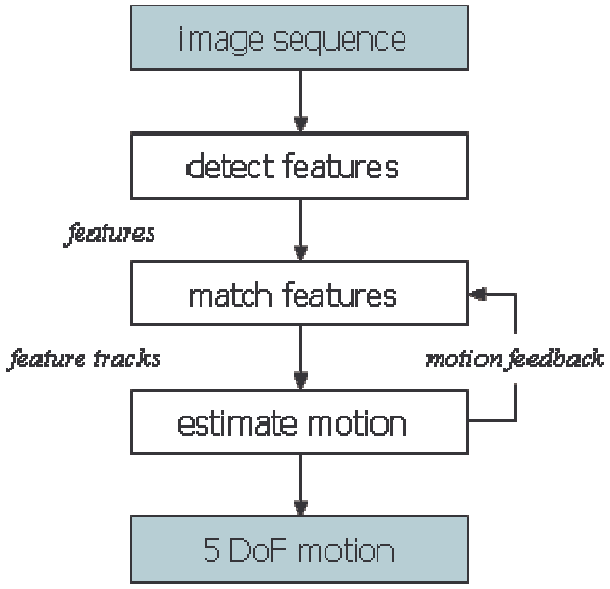


Figure 4.1. Block diagram for motion estimation from a video sequence.

4.2 Feature Detection and Matching

The first step in any algorithm for estimating camera movement from consecutive images in a sequence is the extraction of features from the images and identifying which features in each images correspond to one another. Since the features need to be matched as uniquely as possible, a qualitative definition of a good feature is a region in the image that has strong variations in every direction.

Feature extraction has been investigated in many forms, and many proven algorithms for extracting significant features exists. Of these, the most common are the texture-based features such as the Harris corner detector (Harris & Stephens, 1988), the Shi-Tomasi feature detector and tracker (Shi & Tomasi, 1994), and the affine invariant feature detector (Schmid et al., 2000). These algorithms are based on the directional derivatives of a small region centered about a pixel, and have been proven to be both robust and efficient.

For simplicity's sake, and because it has proven to be a robust outdoor feature detection algorithm (Schmid et al., 1998) , we have chosen to use the Harris corner detector for our implementation of the motion estimation algorithm. In essence, the Harris feature uses the directional gradients of the image (smoothed to reduce the effects of noise) as inputs to an arithmetic calculation to identify significant points of interest. Let $\boldsymbol{I}_x$, and $\boldsymbol{I}_y$ be the directional image gradients at a given point $\boldsymbol{p}$. Then, the Harris feature at $\boldsymbol{p}$ is defined as

$$c(\boldsymbol{p}) = \frac{I_x^2 I_y^2 - (I_x I_y)^2}{I_x^2 + I_y^2}. \tag{4.1}$$

Features are considered to be significant interest points if their value is higher than a certain percent of the rest of the population, typically chosen to be in the range of the 99-99.99 percentile depending on the overall textured-ness of the image. To avoid clustering of feature points about a single image structure, a local mask is applied to pick only the most significant feature for a single image structure.

The next step in the motion estimation algorithm is to locate the position of the features detected in the first image in the second image. This is known as feature matching, or tracking. Like feature detection, there have been many methods proposed for feature tracking. These methods can be broken down into two main groups: correlation methods and optical flow methods. Correlation methods are simple to implement and are appropriate when the inter-frame motion is expected to be greater than a few pixels. Optical flow methods (Lucas & Kanade, 1981; Shi & Tomasi, 1994) work well on small motions and require fewer computations (in general) than correlation methods, in general. For our application, we expect that the interframe motion will be larger than a few pixels, so we will use intensity correlation for the feature matching.

The execution time of the feature matching process is dependent both on the number of features being tracked, the size of the correlation window, and the number of features detected in the second image. To reduce the potential calculations, we have chosen to use a small correlation window (3x3) and to limit the number of correlations by only investigating those features in the second image that are within 50 pixels of the feature in the first image that is being tracked.

4.3 False Match (Outlier) Rejection

No matter the algorithm used for determining correspondences between successive views, the feature matching task is difficult enough that the output will contain false matches along with the true. These false matches are not consistent with the scene's epipolar geometry, and are known as 'outliers', while those true feature matches are known as 'inliers'. For naïve motion estimation algorithms, such as the least-squares algorithm (Hartley, 1995), even a single outlier can cause problems. A typical set of correspondences may contain over 50% false matches, which causes naïve estimators to fail completely.

4.3.1 Previous Efforts

Much work has been done on outlier rejection for epipolar geometry estimation. These techniques – designed to increase the robustness of the epipolar geometry estimation – include

robust geometry estimators as well as pre-filtering techniques. The effects of outliers on the epipolar geometry estimation can be alleviated through robust algorithms (Zhang, 1998; Pollefeys, et al., 1999; Salvi, et al., 2001). However, the computational cost of these algorithms greatly increases with the percentage of outliers present in the data, as seen in Figure 4.2.

Shi and Tomasi (Shi & Tomasi, 1994) proposed a feature tracker that discards matches based on their image residual. Tommasini et al. (Tommasini et al., 1998) extended this concept to include robustness to illumination changes. In addition, to the robustness to illumination changes, Fusiello, et al. (Fusiello et al., 1999) also developed an outlier rejection method that is linked to their feature tracker. Feature matches are rejected if their computed image residual is greater than a certain threshold. Chua et al. (Chua, et al., 2000) compute the epipolar parameters using all available feature matches, filtering out those that do not behave consistently across a triplet of images.

A typical approach to reducing the number of outliers is to limit the set of putative correspondences by only keeping those features that are *mutual best matches* (Hartley & Zisserman, 2000; Pollefeys, et al., 2000; Corke, 2004) – also known as left-right consistency checking to those in the stereo field. In other words, a feature pair $<f_1, f_2>$ is only considered as a potential correspondence if f_2 is the best match for feature f_1 in image 2, and f_1 is the best match for feature f_2 in image 1.

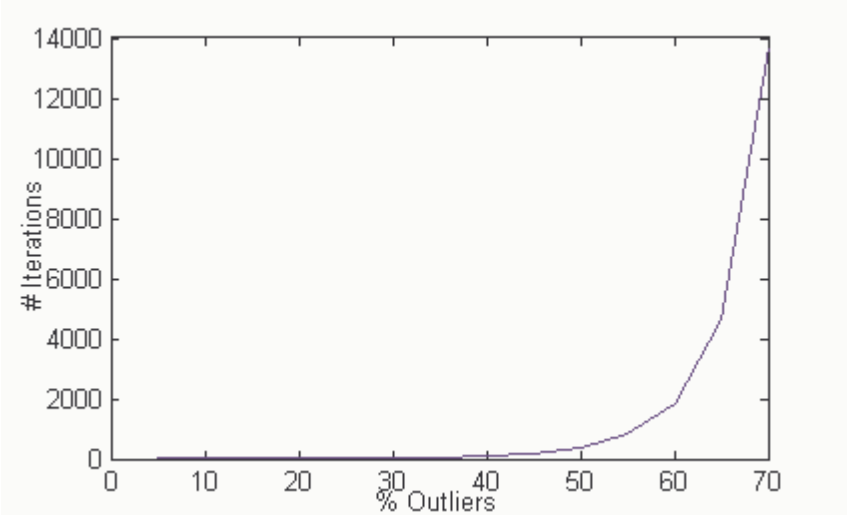


Figure 4.2. Computational complexity of RANSAC, based on the percentage of outliers (false matches) present in the data.

Figure 4.3 shows this process visually. In the left image, a feature P is selected for matching – in this case the upper right corner of the computer monitor. A search region in the right image (shown in yellow) is defined and each feature in the right image that lies inside that search region is compared with the original feature in image 1, typically using an intensity correlation. The feature N that has the highest correlation with P is chosen as a potential correspondence. When the reverse process is performed, if P is the best match for N, then <P,N> is stored as a corresponding pair – a match.

The mutual best match approach greatly reduces the number of outliers present in a set of feature correspondences. However, the set of correspondences generated still contains a significant amount of outliers. In addition, the very nature of the mutual best match approach means that an intensity correlation is done – at minimum – twice for every feature.

4.3.2 Outlier Rejection By Oriented Tracks

In our application, the sensor package is mounted on a mobile platform – either a vehicle, or a 2 driving-, 2 steering-wheeled robot – such that the recording camera's optical axis is orthogonal to the platform's main direction of motion. We assume that the frame rate of the

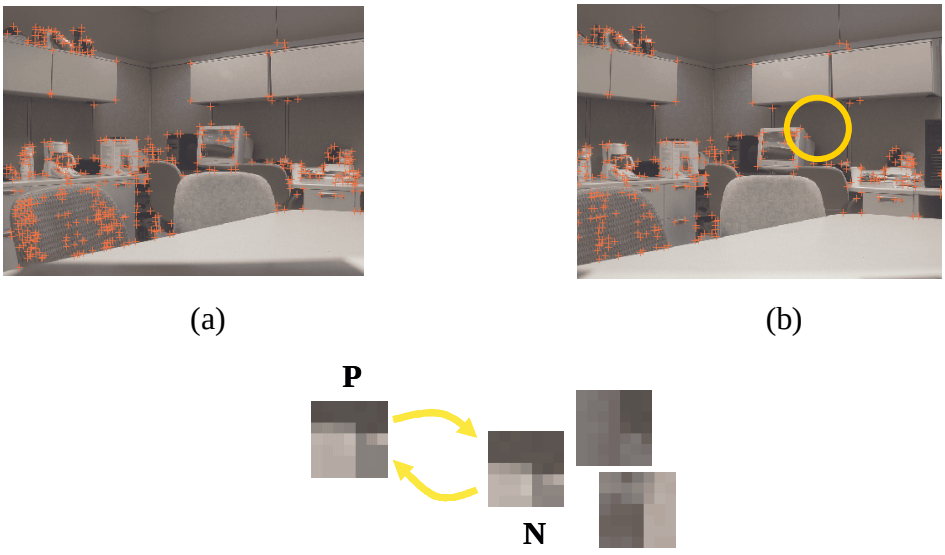


Figure 4.3. Visual explanation of the mutual best match filtering method. (a) The extracted features in Image 1 are marked in red, with the current feature (the one we are trying to find a match for) marked in white. (b) The extracted features in Image 2 are marked in red, with a search region being marked by the yellow circle. (c) The 5x5 regions around the target feature, and all of the features in Image 2 that lie within the search region. The two features <f1,f2> are considered a corresponding pair because they are each the other's closest match within the search region.

camera is high enough so that inter-frame motion is relatively small (<50 pixels). We also assume that the vehicle's motion will be restricted by the environment such that motion along or about the camera's optical axis will be relatively small. For a static scene under these conditions, correspondences that are consistent with the epipolar geometry of the scene will have similar behavior with regards to their motion trajectories – the lines connecting the feature locations in the two images – also called feature tracks.

Figure 4.4b shows the feature tracks for a typical sequence. The feature tracks display motion that is consistent with the epipolar geometry as well as trajectories due to mismatches that are not consistent with the camera motion. For example, in the center of the image are motion tracks that are nearly vertical due to mismatches in the intensity correlation process. In effect, feature pairs of this type are the outliers that limit the efficacy of standard epipolar geometry estimation algorithms. By filtering out all these outlying feature tracks, we can generate a data set similar to that shown in Figure 4.4c. In doing so, we will have reduced the input to a smaller set of feature correspondences whose track orientations are consistent with the epipolar geometry – which will increase the efficacy of the epipolar geometry estimation algorithm.

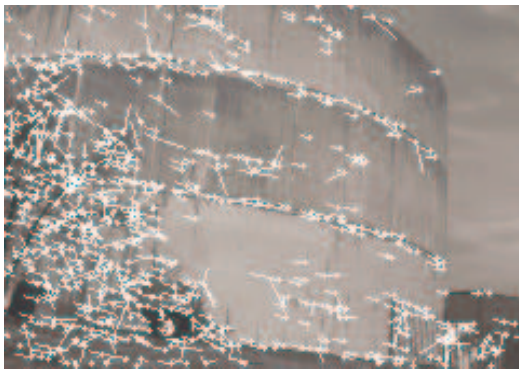
Adam, et al. (Adam et al., 2001) proposed an algorithm – Rejection of Outliers by Rotations (ROR) – to reject false matches from any two views of the same scene that involves virtual rotations of one of the images to identify some common behavior among the correct feature matches. The basic concept is that under a correct rotation, the inliers will have a common feature track direction, and the outliers will be "shaken away". A user-defined number of random rotations are performed and for each rotation, the algorithm looks for feature tracks that lie in a common direction. The correct feature tracks are those that lie within a specified angular distance from the common direction, averaged over the best rotations. This algorithm has proven to be effective when applied to general motion viewing a static scene.

Under our application, the motion constraints of the mobile sensor platform mean that the correct matches will share a common behavior in their feature tracks for the original image pair, with no rotations needed. With this knowledge, we have designed an algorithm similar to ROR, that requires no virtual rotations of the images and relies only on a single user-specified parameter, as opposed to the five required for the ROR algorithm. Our proposed algorithm is outlined as follows:

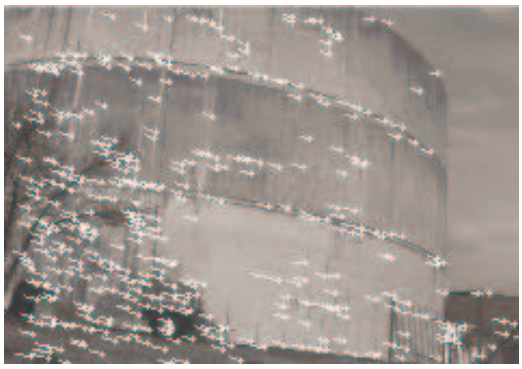
Oriented Tracks Algorithm

- Begin with a set of potential correspondences $c_i = (a_i, b_i)$ between images A and B.
- Calculate each feature track direction from the line joining the locations of the corresponding pair
- Estimate the probability distribution of the feature track directions.
- Find the main orientation as the mode of the distribution of feature track directions m.
- Find the angular distance d_i from each feature track to the mode m.
- A feature track is considered consistent with the main orientation if for some acceptance region ε, $d_i < \varepsilon$.

(a)



(b)



(c)

Figure 4.4. Example of track orientations. (a) First image in a video sequence. (b) Second image, showing matched features and their respective track orientations. (c) The same image showing only those tracks with orientation consistent with the epipolar geometry.

4.3.3 Estimating The Probability Distribution Of The Track Angles

There are many ways to estimate a probability density function of a random variable. Generating a histogram of the data can give a quick estimate of the distribution, but is subject to errors due to the selection of bin width and interval, and is mathematically unsuitable for a variety of analysis tasks. A better method for estimating the probability density function is to use a kernel density estimator.

The kernel density estimator uses a weighted summation of kernel functions, placed at the observation data, to estimate the density. These kernels can be chosen to have all of the desired mathematical properties (e.g., continuous, differentiable, etc.) and their parameters can be chosen to fit the data in some optimal sense (Silverman, 1986). The kernel estimator with a given kernel K is defined by

$$\hat{pdf}(x) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x - X_i}{h}), \tag{4.2}$$

where h is the kernel width, n is the number of sampled data points, and X_i is the ith observation of the random variable X. As the kernel width tends to zero the estimated density becomes a sum of delta functions – the under-smoothed case. As the width becomes large the probability density function is over-smoothed and all detail is obscured.

Figure 4.5 shows the comparison of the probability density estimation of the feature track directions from Figure 4.4. A standard histogram (Figure 4.5a) is shown beside density functions determined from a weighted sum of Gaussian kernels. Figure 4.5b shows the case where the kernel width is determined empirically, while in Figure 4.5c the kernel width is optimally chosen using an L-Stage Plug In method (Wand & Jones, 1995).

Recall that we are interested in the main orientation as defined by the feature track direction that has the greatest probability. Thus, we are more interested in the location of the PDFs maximum than its shape. We experimented with a number of kernel width optimization algorithms – e.g., L-Stage Plug-In, Empirical, Bootstrap – and the results showed that for every case, the maxima occurred at the same location – the main orientation. Thus, any of the above kernel width optimization procedures will do.

For our work we have chosen to use the L-Stage Plug In method to estimate the bandwidth of our Gaussian kernel function. This method was chosen for its ease of implementation and its asymptotic minimization of the Mean Integrated Square Error (MISE) between the estimated probability density and the actual density function. The MISE can be approximated as

$$AMISE(\hat{f}) = \frac{R(K)}{nh} + \frac{1}{4}h^4 \mu_2(K)^2 R(f''), \tag{4.3}$$

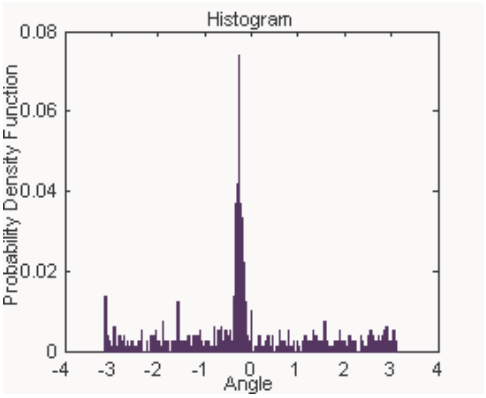
where $R(K)$ and $\mu_2(K)$ are constants that depend on the selection of kernel function K, and $R(f'')$ is a constant depending on the unknown density function of the random variable.

The optimal bandwidth wrt. this criteria has the closed form solution

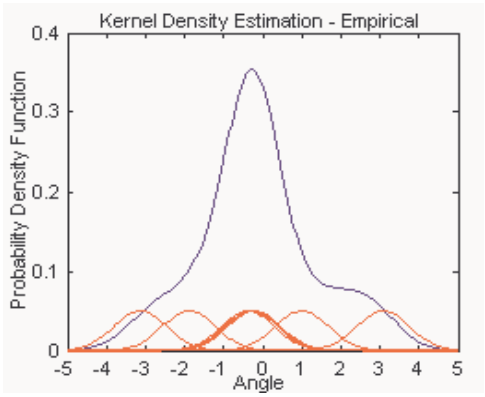
$$h_{opt} = \left[\frac{R(K)}{\mu_2(K)^2 R(f'')n}\right]^{1/5}. \tag{4.4}$$

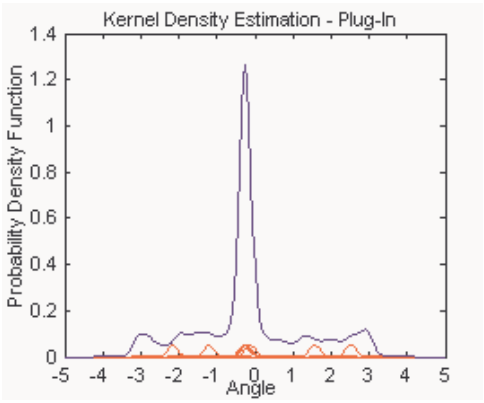
Silverman's solution for the optimal bandwidth assumes that the true distribution function can be represented by a normal distribution. Using this assumption in combination with a Gaussian kernel, the optimal bandwidth for kernel density estimation can be computed as

$$h_{opt} = \left[\frac{4\hat{\sigma}^5}{3n}\right]^{1/5}, \tag{4.5}$$



(a)





(b) (c)

Figure 4.5. Comparison of probability density estimation using histograms and kernel density estimators for the feature tracks orientations shown in Figure 4.4b. Notice that for every bin width/kernel width choice and every density estimation algorithm, the maximum density occurs at approximately the same location, -0.25 ± 0.05 radians. (a) Standard histogram – bin width set at 0.0175 radians (1°). (b) Empirical estimate of the density function using a normal kernel with a width of 0.5 radians. (c) Density estimated using an L-Stage Plug In method to optimally choose the kernel width.

where $\hat{\sigma}$ is either the sample standard deviation or the standardized inter-quartile range. In practice the smaller of the two is used.

We also investigated the effect of the kernel width optimizer on the choice of the main orientation. Our experiments showed that the differing kernel widths chosen by the various optimizing algorithms cause the peak of the density function to shift minutely. However, in practice, the acceptance window specified for track consistency is significantly larger than the shift in the maxima location between various estimators. Thus, the choice of methods for density estimation has no real impact on the outlier rejection technique.

4.3.4 Choosing The Acceptance Region

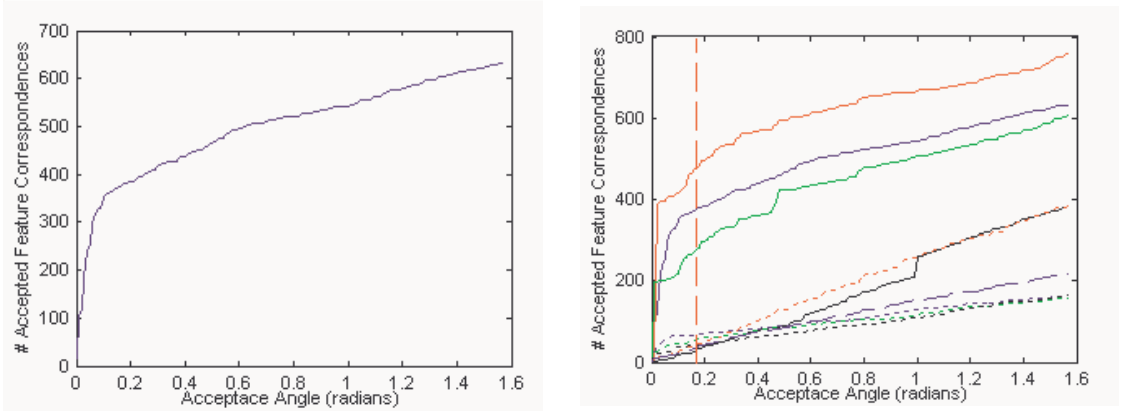
The parameter controlling the width of the "consistent" region is the only user-selected parameter in this outlier rejection method. If this region is too large, feature tracks inconsistent with the epipolar geometry may be kept in the correspondence set. Too small, and the noise inherent in digital acquisition systems will cause consistent feature matches to fall outside the threshold, and thus will remove them from the set of correspondences.

To test the effect the acceptance window width has on the number of track correspondences chosen to be consistent with the main orientation, the window size was allowed to vary from 0 to $\pi/2$ and the number of accepted feature correspondences was recorded. Figure 4.6 shows the results of this experiment. In Figure 4.6a, the results for this experiment are shown for the scene given in Figure 4.4. Figure 4.6b shows the results from a randomly selected group of 10 scenes.

For a clean data set – one whose density function has a sharp peak at the main orientation and a fairly uniform distribution otherwise – such as the scene shown in Figure 4.4, the graph exhibits a sharp increase in accepted feature pairs for a short time, and then the number of additional matches begins to slow down. Figure 4.6a shows this behavior, with a "knee" clearly evident. Figure 4.6b shows that even scenes imaged with a variety of noise (motion blur, resampling, pixelation, etc.) exhibit a knee-type structure.

Taking the acceptance window width to be just above the knee allows us to retain the most consistent feature tracks, while providing for some system error. The vertical line shown in Figure 4.6b represents the window size chosen for our experiments in the next section. It has been empirically determined from our test data to provide the best tradeoff between outlier rejection and maximum inliers for the epipolar geometry estimation procedure. As another option, since the window width choice is used to account for errors in the vision system, it could be predetermined by some foreknowledge of the system noise.

Tracks inconsistent with the main orientation - i.e., those feature tracks whose angular distance from the main orientation does not lie within the acceptance window – are determined to be outliers and are filtered out, with the remaining correspondences retained as our new, concise feature set.



(a) (b)

Figure 4.6. The number of OT correspondences vs. the acceptance window size for our demonstration data sets. The number of correspondence pairs has been normalized to account for variations in the scenes. The dash/dot blue line indicates our empirically selected threshold of acceptance. (a) The curve for the scene shown in Figure 1, showing a clean "knee". (b) The curves for 10 of our test scenes. The vertical line shows the empirically chosen threshold used for the results given in the next section.

4.3.5 Experimental Results For Outlier Rejection

To demonstrate the efficacy of our outlier rejection method, we compared its filtering results to those of the typical mutual best match method for a variety of image pairs taken from our test platform, using a high-definition camcorder (JVC GR-HD1). Figure 4.7 shows three of these test pairs, which contain several effects that cause outliers in the estimation of the epipolar geometry, including: digitization noise, resampling noise, motion blur, parallax effects, glare, shadows, etc. The scenes in Figure 4.7 increase in complexity from top to bottom.

For each image pair, a standard Harris detector was used to select approximately 1000 critical points. Matches between the images were found using a standard intensity correlation method. MBM was used as the baseline outlier rejection method. Feature track orientations were calculated from the Image 1 → Image 2 feature matches, and the main orientation was determined via the method discussed in the previous section, yielding the Oriented Tracks (OT) correspondences. Epipolar geometry estimation was performed as discussed in (Hartley & Zisserman, 2000) and the number of inliers was determined for both the MBM and the OT correspondence sets. For our experiments, inliers are determined as those features that lie within 2 pixels of the computed epipolar line. The procedures for feature detection, outlier rejection, and epipolar geometry estimation were implemented in Matlab and run on a 2.4 GHz Pentium processor.

(a)

(b)

(c)

Figure 4.7. Image pairs used to test the outlier rejection algorithm, taken from a high-definition video sequence. Image titles corresponding to the results shown in Figure 4.8 and Table 4.1 are: (a) Tanks, (b) Intersection, and (c) Courthouse.

In order to evaluate the pre-filtering results, we selected 50 true matches by hand for each image pair. Then, we used RANSAC to robustly estimate the epipolar geometry from these matches, yielding a set of 'true matches' that we then used as ground truth for comparison purposes. A comparison of the results obtained with our outlier rejection method and those obtained with MBM is shown in Table 4.1.

The results show that even for datasets that contain fewer than 40% true matches, our filtering method improved the true/false match ratio to better than 90%. This improvement in the quality of the feature correspondences leads directly to an improvement in the calculation of the epipolar geometry of the scene from feature matching, using RANSAC, in terms of both efficiency (time of execution) and accuracy (more inliers).

The RANSAC procedure used to estimate the epipolar geometry from the filtered set of correspondences does so through a randomly selected sample of feature matches, the choice of which can greatly affect the outcome in terms of iterations and inliers. In order to present an "average performance", we ran the geometry estimation process 1000 times for each test image pair for both MBM and OT filtered correspondences.

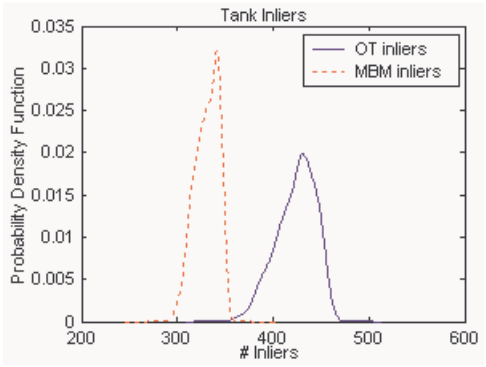
Figure 4.8 shows the results of those trials. For every image pair, a probability density function of the number of inliers was computed, using the same plug-in method discussed previously. Figure 4.8 shows that for every case, our OT method gives a higher number of inliers for the epipolar geometry estimation than the mutual best match approach. In addition, the plots also

Table 4.1. Comparison of OT outlier rejection and MBM on real images.

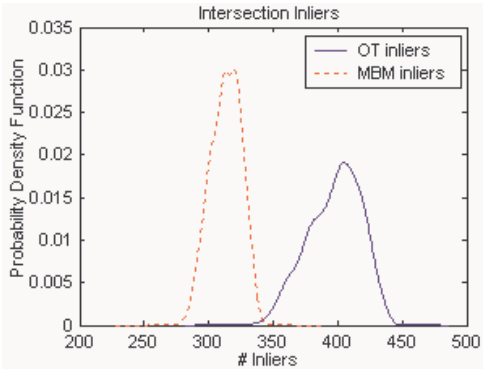
	Tanks		**Intersection**		**Courthouse**	
Total # of Features	910		868		949	
True Matches	463 (51%)		524 (60%)		432 (46%)	
	Oriented Tracks	*Mutual Best Match*	*Oriented Tracks*	*Mutual Best Match*	*Oriented Tracks*	*Mutual Best Match*
Filtered Matches	416	458	503	503	355	435
False Matches Accepted (Misses)	15 (4%)	132 (27%)	23 (5%)	92 (18%)	11 (3%)	120 (28%)
# RANSAC Iterations[1]	6	30	6	16	6	53
Outlier Rejection (s)	0.27	133.69	0.22	131.42	0.34	130.66
RANSAC (s)[2]	0.16	0.77	0.19	0.45	0.13	1.16
Total (s)	0.33	134.46	0.41	131.97	0.47	131.82

[1] Median number of RANSAC iterations taken from 1000 trials.

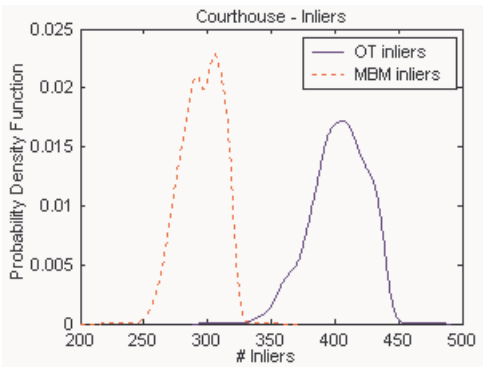
[2] Median time for RANSAC iterations taken from 1000 trials.



(a)



(b)



(c)

Figure 4.8. OT and MBM inlier distributions for the images shown in Figure 4.7.

show that for the most part the distributions are disjoint, which means a substantial improvement in the number of inliers for our algorithm as compared to mutual best match.

In addition to a higher number of inliers, the OT filtered correspondences also yielded fewer RANSAC iterations, providing a performance increase in speed. Table 4.1 shows that increase for our demonstration data sets. By presenting the geometry estimation algorithm a better set of correspondences, we reduce the number of iterations by at least half, and often the improvement is much greater – on the order of 1/10. This decrease in iterations corresponds to a decrease in the amount of time the geometry estimation process takes. Overall, because we deliberately avoid duplicating the intensity correlation process – required by MBM – the amount of total time for pre-filtering and epipolar geometry estimation for our method is a mere fraction of that necessary for MBM. In total, our OT method for outlier rejection reduces the computation time by 2 minutes (a speedup of over 500 times) *per image pair*, which is a significant savings when the goal is to process an entire video sequence.

These results show that under the assumed motion constraints, our oriented tracks outlier rejection algorithm outperforms the mutual best match method in terms of reduced computations (improved speed) and has a more accurate epipolar geometry estimation – as ascertained by a greater number of inliers for the epipolar geometry estimation.

4.4 Pose Estimation

The epipolar geometry of an image pair is the projective geometry between the pairs. It is independent of scene structure, and depends solely on the camera's internal parameters and the relative change in pose between the two views. Thus, a point in one image, combined with the camera parameters, defines an epipolar line in the other view on which the corresponding point lies. The epipolar geometry is represented by the *fundamental matrix* **F,** a 3x3 matrix of rank 2 with 7 degrees of freedom.

If ***x*** is a given feature in the first image and ***x'*** is the same feature in the second image, then the epipolar constraint can be written as $\boldsymbol{x}\mathbf{F}\boldsymbol{x'} = 0$. Thus, we can calculate the fundamental matrix **F** with a minimum of seven pairs of feature correspondences. In general, however, we have more than the minimum number of correspondences, leading to an over-constrained system. This system can be solved either using singular value decomposition (SVD), or through a robust method such as RANSAC (Zhang & Zhao, 1995; Pollefeys, et al., 1999; Hartley & Zisserman, 2000). Due to the inaccuracies of even the best feature matching methods, we have chosen to use the RANSAC method for its robustness to noise.

Using the calculated **F** and the *a priori* knowledge of the camera's intrinsic parameters ***K***, we can calculate the *essential matrix* **E** – a special version of the fundamental matrix, for normalized image coordinates – which has the form

$$\boldsymbol{E} = \boldsymbol{K}'^{T}\boldsymbol{F}\boldsymbol{K} = [\boldsymbol{t}]_x \boldsymbol{R}, \tag{4.6}$$

where ***t*** and ***R*** are the translation and rotation, respectively, of the camera between the two views, and $[\boldsymbol{t}]_x$ is the skew-symmetric matrix

$$\left[(t_x, t_y, t_z)^T\right]_x = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}. \tag{4.7}$$

The translation vector ***t*** can be shown to be the solution to $\min\left\|\boldsymbol{E}^T\boldsymbol{t}\right\|$, which is the unit eigenvector with the smallest eigenvalue for the matrix $\boldsymbol{E}\boldsymbol{E}^T$. Rearranging (4.6) results in $R^T[-t]_x = E^T$. A solution for ***R*** in the presence of noise can be found by solving $\min\left\|(R^T[-t]_x - E^T)\right\|$.

5 Automated Spatial Resolution Improvement Using Kriging

Kriging is a procedure for data interpolation that makes use of regionalized variables – a cross between random variables and deterministic processes – to define a stochastic process model, under which the interpolation is done. Thus, kriging estimates the values at specified locations, using observed data to drive the process, optimized with respect to specific error criteria. This error criteria is the squared prediction error at the unobserved locations. The measured data provide the 'support' for the estimated values, and the quality, size, shape, and orientation of the observed values influence the ability to accurately predict the unknown sample values.

Kriging has advantages over the previously mentioned interpolation methods in that:

- It makes no assumptions about the type of data being estimated
- It makes no assumptions about the sampling criteria, density, shape, orientation, etc.
- It has a built-in criteria for goodness of fit
- It has been proven to outperform all other interpolation methods – under specific conditions (e.g., when the relationship between the data can be readily modeled by a parametric function) – and not to perform worse
- Ideal for non-uniform, sparsely sampled data

The overall process for kriging consists of 3 steps:

- Estimate the spatial correlation between the measured samples
- Construct an ideal model that best fits the estimated spatial correlation
- Estimate the new surface values using kriging

5.1 Estimating The Spatial Correlation

The estimation of the spatial correlation of surface data in regionalized variable theory is typically done with the use of variograms and the semivariance – a measure of the intrinsic relationship between measured points. The variogram is a function which characterizes the dependence of data points measured in a region of interest. The semivariance is simply half the variance of the differences between all measured points spaced an equal distance apart.

The magnitude of the semivariance between two points is proportional to the distance between them. Points close together yield a smaller semivariance than points that are farther apart. Thus, the semivariance at distance d=0 would be zero, as there is no difference between a point and itself. As the distance increases, so does the semivariance, until a saturation point is reached, where the expectation that a point at distances further than this threshold no longer affects the specified data point. This distance is called the range of the regionalized variable. This range – derived from the data itself – is the *de facto* maximum neighborhood size for estimating a new surface value. The plot of the semivariances with respect to distance is known as the variogram (semivariogram).

The variogram is a visual and functional measure of how quickly the surface changes, on average. The variogram is a positive function, as all of its elements are greater then or equal to zero. It should also be noted that the variogram can be directionally dependant, according to the anisotropic nature of the measured data.

In the simple case of regularly sampled data, the computation of the semivariance is quite straightforward. Assuming the sampling interval (lag) is d, the semivariance for distances equal to multiples of d can be computed as

$$\gamma(d) = \frac{1}{2N(d)} \sum_{i=1}^{N(d)} (z_i - z_{i+h})^2 , \qquad (5.1)$$

where z_i is the measurement of a regionalized variable taken at location i, z_{i+h} is another measurement taken h intervals away, $N(d)$ is the number of points used per lag interval.

Once the experimental variogram has been calculated, an ideal parametric model is fit to the data through an automatic optimization method. Least-squares fitting of a number of ideal models is performed, and the one with the best match to the data is used as the ideal variogram model for the kriging process. An ideal parametric model is used to simplify the estimation process and to increase the robustness to measurement errors.

When the data is not omnidirectional – i.e., it has a significant directional trend – directional variograms can be used to weight the values more for samples that lie along the directional trend. In this case, the h used in (5.1) becomes a vector, rather than a simple distance, and the semivariance is calculated exactly as before. Typically, since having a significant number of data points separated by exactly the same vector is not very likely, the set of all possible directions is partitioned into a set of classes, as seen in Figure 5.1. Separation

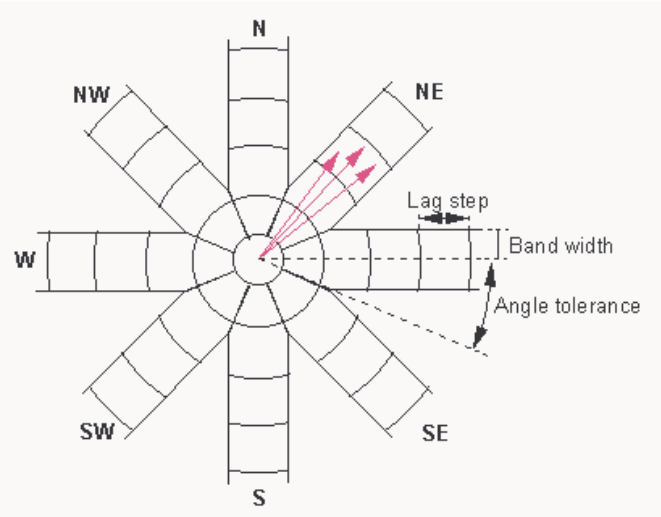


Figure 5.1. Directional variograms (Davis, 1986).

vectors are hashed into these classes, and the semivariance is calculated for all the data pairs that lie in each class.

5.1.1 Estimation Of New Values By Kriging

Kriging is the actual process of using the parametric variogram model to estimate the surface value at the specified location. Earlier we mentioned that kriging outperforms all other methods of surface interpolation. This is the case when the data is stationary and the proper form of the variogram has been selected. The most common form of kriging used in engineering applications is punctual (point) kriging – where the estimate for a single point is calculated from the values of nearby points, as opposed to block kriging where the data is processed in sets of points.

In punctual kriging, the estimate of an unknown surface vale uses a weighted summation of other nearby known points:

$$Z_e(p) = \sum w_i Z(p_i). \tag{5.2}$$

The error associated with this estimate Z_e and the actual value Z_a of the surface at this location is

$$\varepsilon_p = Z_e(p) - Z_a(p). \tag{5.3}$$

Ideally, kriging attempts to minimize this error. The variance of this error is the amount of scattering of the estimates about their true values

$$\sigma_z^2 = \frac{\sum_{i=1}^{n} [Z_e(p_i) - Z_a(p_i)]^2}{n}. \tag{5.4}$$

Obviously, the estimation and its error are dependent on the weights chosen in (5.2). Optimal weights, therefore, would be those that produce the minimum estimation variance. These are found by solving a system of equations consisting of the weighted semivariances between measured points, and the estimated semivariances between the unknown point and the known values.

$$\begin{bmatrix} \gamma(d_{11}) & \cdots & \gamma(d_{1n}) \\ \vdots & \ddots & \vdots \\ \gamma(d_{n1}) & \cdots & \gamma(d_{nn}) \end{bmatrix} \begin{bmatrix} w_i \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} \gamma(d_{1p}) \\ \vdots \\ \gamma(d_{np}) \end{bmatrix}, \tag{5.5}$$

As a simple example, let us estimate an unknown value $Z_e(p)$, using the known values *Z1*, *Z2*, *Z3*, and *Z4*. Since we have 4 points that will contribute to the estimation, 4 weights must be determined. Thus, we have 4 simultaneous equations:

$$\begin{aligned} w_1\gamma(d_{11}) + w_2\gamma(d_{12}) + w_3\gamma(d_{13}) + w_4\gamma(d_{14}) &= \gamma(d_{1p}) \\ w_1\gamma(d_{21}) + w_2\gamma(d_{22}) + w_3\gamma(d_{23}) + w_4\gamma(d_{24}) &= \gamma(d_{2p}) \\ w_1\gamma(d_{31}) + w_2\gamma(d_{32}) + w_3\gamma(d_{33}) + w_4\gamma(d_{34}) &= \gamma(d_{3p}) \\ w_1\gamma(d_{41}) + w_2\gamma(d_{42}) + w_3\gamma(d_{43}) + w_4\gamma(d_{44}) &= \gamma(d_{4p}) \end{aligned}, \tag{5.6}$$

where $\gamma(d_{ij})$ is the semivariance between points i and j, and d_{ij} is the distance between the two points. The semivariance values are taken from the parameterized variogram.

To assure that the solution is unbiased, a further constraint of $\sum w_i = 1$ is usually applied. This leads to an overconstrained system, so another variable is added to the system, called the Lagrangian multiplier, to insure a minimum error solution is obtained. Thus, the system of equations shown in (5.6) is more commonly solved as:

$$\begin{aligned} w_1\gamma(d_{11}) + w_2\gamma(d_{12}) + w_3\gamma(d_{13}) + w_4\gamma(d_{14}) + \lambda &= \gamma(d_{1p}) \\ w_1\gamma(d_{21}) + w_2\gamma(d_{22}) + w_3\gamma(d_{23}) + w_4\gamma(d_{24}) + \lambda &= \gamma(d_{2p}) \\ w_1\gamma(d_{31}) + w_2\gamma(d_{32}) + w_3\gamma(d_{33}) + w_4\gamma(d_{34}) + \lambda &= \gamma(d_{3p}) \\ w_1\gamma(d_{41}) + w_2\gamma(d_{42}) + w_3\gamma(d_{43}) + w_4\gamma(d_{44}) + \lambda &= \gamma(d_{4p}) \\ w_1 + w_2 + w_3 + w_4 &= 1 \end{aligned}. \tag{5.7}$$

The weights that are the solution of this system are then plugged into (5.2) to estimate the surface value for the point of interest. Thus, in the general form, the kriging equations are

$$\begin{bmatrix} \Gamma_z & \mathbf{1} \\ \mathbf{1} & \mathbf{0} \end{bmatrix} \begin{bmatrix} w \\ \lambda \end{bmatrix} = \begin{bmatrix} c_z \\ \mathbf{1} \end{bmatrix}, \tag{5.8}$$

where Γ_z is the semi-variance matrix taken from the semi-variogram, c_z is a vector of the observed values used for kriging, w is the solution for the weights of the ordinary kriging estimator, and λ is the Lagrange multiplier.

5.2 Parameter Selection

The advantages of kriging are twofold: (1) it provides estimates of the surface values at unknown locations with a minimum *quantifiable* error, and (2) in its natural form it is a completely data-driven approach. The only user-specified parameters are the sampling interval (lag) for measurement and the library of variogram functions provided to work from. The experimental semivariances can be fit to the "best match" variogram in the library through standard fitting methods – e.g., least squares matching.

However, kriging does have some disadvantages that can be overcome with prior knowledge of the system, and the addition of some user-specified constraints. Kriging in its native form typically over smoothes the estimates, based on the proportion of the surface variance that is contributed from noise. The noisier the data, the less the samples truly represent their immediate vicinity. The addition of the Lagrangian multiplier and the unit summary of weights improves the system's robustness to noise, but does not allow for better estimates of the surface.

Regularization techniques are typically used in data fitting approaches to provide a user-specified tradeoff between data fidelity and surface smoothness. Typically, they provide a parameter – the regularization parameter – that specifies the degree to which the measured values are assumed to match with the true surface. This parameter can be experimentally determined for specific scenarios, or can be algorithmically determined through such methods as Total Variation. One of the goals of this project is to develop a criteria that would allow the sharp edges inherent to our 3D regions of interest to be preserved, while maintaining an optimal error variance for the overall data set (locally).

Additionally, because our surfaces are locally significant, but change over distance, it is typical that the full range of the kriging estimators may not accurately represent the local surface. An example would be a sign on a building, reading "Chinese Supermarket". Over the span of the data, points taken on the 't' might exhibit some effect on the 'C', simply because they are the same height from the building wall. However, it should be obvious from an English language standpoint that the 't' would have no effect whatsoever on the 'C', as any combination of letters could follow. Therefore, restricting the range of influence for

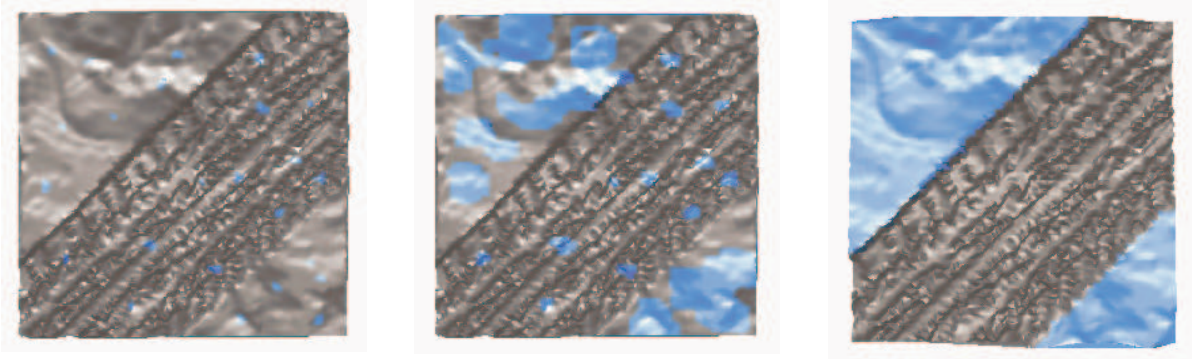
the data points used to perform the kriging can indeed improve the overall estimate of the surface values.

The final "parameter" that needs to be tuned for optimal, automatic performance of the detail enhancement procedure is the selection of the regions of interest on which the kriging is to be performed. Work has been done using surface curvature to define areas of interest for smoothing applications. It is not unreasonable to believe that we can use similar techniques to identify those regions in our scanned data that have geometrical significance, as a precursor to the local resolution enhancement.

5.2.1 Segmentation of the Regions of Interest

In the applications we have observed, the Regions of Interest (ROIs) chosen for geometry refinement typically stand out from the rest of the data – in terms of spatial frequency, curvature variation, and shape – and tend to be much smaller than the overall dimensions of the entire dataset. Segmentation of these regions can be done manually, if the application allows, but automatic segmentation of these regions requires a mathematical or heuristic description of what makes these regions – which are easily picked out by a human observer – different from the rest of the data. For our method, we have chosen to use curvature as the indicator for ROIs. We follow the watershed-based approaches of Page et al. (2003) and Sun et al. (2002) In our case, we use the surface curvature as the marker function for the region growing and merging segmentation. Areas of high curvature variation fall out in the segmentation process, and are thus identified as our ROIs. This process is demonstrated in Figure 5.2.

The first step in the segmentation process is the selection of seed triangles. We randomly choose a small percentage – roughly 2% for the datasets shown here – of the total number of



(a) (b) (c)

Figure 5.2. Segmentation process, showing the (a) seeding, (b) region growing and merging, and (c) small region culling.

triangles in the dataset, without allowing any of the seed triangles to be coincident. Figure 5.2a shows the highlighted seed triangles for our example dataset. Next is the region growing procedure. The seed patches are enlarged by assimilating coincidental triangles if their curvatures are similar. Patches that contact each other are merged, providing that their overall curvatures are similar. After several iterations of growing and merging, the number of distinct patches is reduced, and there is a mixture of both large and small triangle count patches, as seen in Figure 5.2b. The region growing and merging process ends when no more triangles/patches can be merged together based on common curvatures.

After the region growing and merging is complete, a patch culling algorithm is used to remove those regions with fewer than a specified number of triangles. For the results shown in this paper, we used a cutoff threshold of 25 triangles. This number was empirically chosen, but could be automatically selected from the data – a percentage of the triangles in the largest patch, for example. What remains is a set of large surface patches that have a relatively constant curvature, as seen in Figure 5.2c. Taking the inverse selection yields regions of high curvature changes, and thus our target ROIs.

The segmented ROIs are used both as the domain for geometry refinement of the surface, as well as the training data for the kriging process. Experiments have shown that using the data of the ROIs only – rather than that from the entire dataset – to build the variogram model yields higher accuracies in the estimates of the new data within the ROIs. More discussion of these experiments and their results will be given in the next section. By restricting the training data to lie only in the ROIs, we get an improvement in surface fit due to the removal of the influence of the data that is "not of interest".

5.2.2 Selection of Variogram Parameters

The selection of the parametric variogram model used to represent the calculated semivariances is another area which needs to be automated for a completely hands-off geometry refinement algorithm. Variograms are usually selected from a library of positive definite functions and manually fit to the estimated semivariances. In order to automate this process, we perform a least-squares fitting of a selection of functions, and choose the function that best fits the estimate. The functions evaluated as possible models for the variogram include spherical, exponential, Gaussian, and Matern:

Spherical model:

$$\gamma(d) = \begin{cases} c_s\left(\frac{3d}{2r} - \frac{d^3}{2r^3}\right), & d \le r \\ c_s, & d > r \end{cases}$$

Exponential model:

$$\gamma(d) = c_s\left(1 - e^{-3d/r}\right)$$

Gaussian model $\gamma(d) = c_s\left(1 - e^{-\left(3d/r\right)^2}\right)$

Matern model: $\gamma(d) = c_s\left[1 - \frac{1}{2^{\alpha-1}\Gamma(\alpha)}\left(\frac{d}{r}\right)^{\alpha} K_{\alpha}\left(\frac{d}{r}\right)\right]$

In these models c_s is the sill of the variogram, r is the range, Γ is the gamma function, and K_α is a Bessel function.

The approach used to fit the variogram models is similar to the weighted least squares approach described by Cressie (1985). The method is designed to give higher emphasis to data pairs lying close in space, as well as to lags associated with many pairs of data points. This allows for the automated selection of variogram parameters. This fitting is constrained to best fit the function to points evaluated near the source, since for our intended applications we are interested only in the best local fit of the data.

5.2.3 Regularization to Account for Microscale Variation

For real world experimental systems, data sampled at one location over and over incurs variation in the sample. Such variations are caused by system noise, transient delays, environmental changes, etc. In order to account for variance at a lag of 0, traditional kriging places a nugget effect at the origin of the variogram. This essentially acts like a DC component, shifting the entire variogram upwards. However, is this truly appropriate for a system where the measurement variation in different regions is likely to change? To accommodate for the regional variational differences, we employ a regularization procedure that models the micro-scale variations independently. This regularization approach alleviates the effect of local measurement errors and can be tuned to reduce the smoothing effect that the nugget parameter introduces and that is common to most kriging implementations, as discussed by Zinger at al (2002).

Given that Equation (5.5) can be written as $Ax=b$, we can solve the system using Ordinary Least Squares

$$x = (A^T A)^{-1} A^T b\,. \tag{5.9}$$

This is equivalent to zero-order Tikhonov regularization. Since $(A^TA)^{-1}$ is ill-conditioned, we introduce a regularization parameter λ to increase stability of the system

$$x = (A^T A + \lambda I)^{-1} A^T b\,. \tag{5.10}$$

λ can be chosen to be either a scalar or a vector. It can be automatically determined through validation methods such as autocorrelation, or empirically chosen through experimentation. We suggest choosing a λ that reflects the amount of uncertainty present in the data

acquisition system. For the experimental results given in the next section, λ was chosen using the L-curve analysis technique of Hansen (1998).

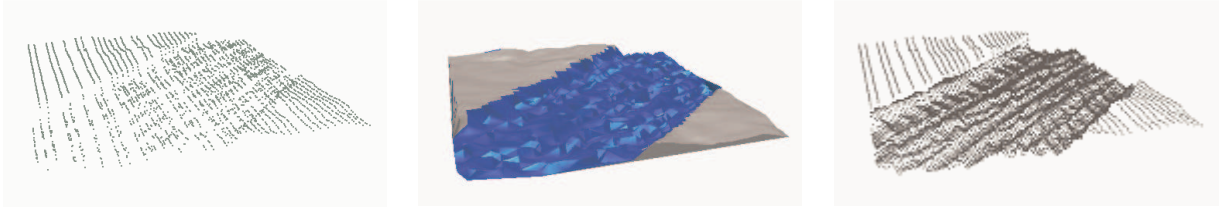
5.3 Geometry Spatial Resolution Enhancement Results

We have implemented the basic version of automated kriging in Matlab, and have tested the implementation on a variety of datasets, including synthetic data, data collected using our Mobile Scanning System, and real data acquired elsewhere. The implementation can accommodate both isotropic and anisotropic data input. The processing flow of the geometry enhancement algorithm is: (1) read in the data, (2) set the ROIs for processing (either manually or automatically), (3) develop the model of the underlying spatio-statisical characteristics of the data, and (4) estimate the surface values at the new grid location.

As an example, see Figure 5.3 where the sparse and unevenly sampled surface of a tire tread impression is resolution enhanced. The tire impression is recognizably different from the ground surface surrounding it and is segmented for processing. A statistical model of the underlying surface is built from the data samples, and then the 3D surface is enhanced. In this example, the dataset represents a small portion of a larger terrain dataset, encompassing perhaps a kilometer's worth of data.

To demonstrate the efficacy of any geometry enhancement algorithm, there needs to be both a qualitative and quantitative check on the output. To this end, we have developed a piece of software that can simulate the performance of a mobile scanning system on a given 3D model. Our Mobile Scanning Simulator utilizes a dense 3D model as a ground truth input and a known sampling scheme to simulate the 3D data acquisition process. Using the resulting data scans, we can perform geometry enhancement with any given algorithm and compare the results back to the original dense model. This gives us a check to compare how good the final results actually are, both visually and metrically.

The simulator, written in Visual C++, allows the user full control over the "data acquisition" parameters, including the density of the scanline, the assumed path of the scanning system,



(a) (b) (c)

Figure 5.3. Geometry enhancement procedure. A sparsely sampled surface (a) has a region of interest (the tread impression) that is segmented (b) and then enhanced (c).

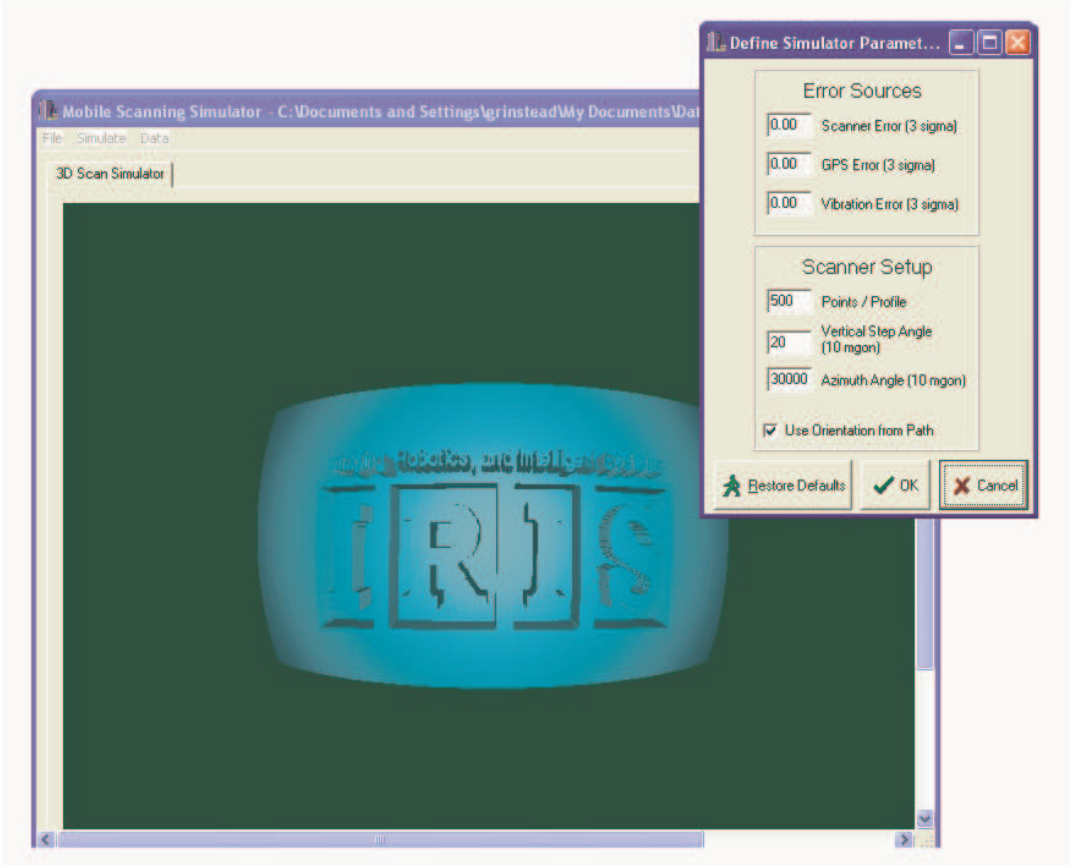


Figure 5.4. Screenshot of the Mobile Scanning Simulator developed for the analysis of the results from our detail-enhancement algorithm. The input model is an ideal surface, to which the processed results will be compared.

the noise models of the system, and the relative uncertainty of each stage of pose estimates. A screenshot of the simulator can be seen in Figure 5.4, with a 3D model of the IRIS logo embossed on a spherical surface.

This simulator has been used to validate the results of our automated kriging spatial enhancement method for both synthetic and real data. The results from the experimentations performed using the Mobile Scanning Simulator can bee seen in the following sections.

5.3.1 Experiments Using Synthetic Data

The synthetic dataset chosen to demonstrate the performance of this method is a model of tire/soil interaction. In this example, you can imagine that a vehicle has recently traveled over loose soil, leaving an impression in the soil. A 3D scanner was then used to capture a

3D model of the impression. In this case, a synthetic model was created by taking a portion of a Digital Elevation Model (DEM) and scaling it to provide a realistic underlying ground surface. A tire tread design was then embossed on the surface to provide a high-density, realistic model of a tire tread impression. The resultant model can be seen in Figure 5.5 and is labeled as "Reference".

The mixture of sharp features and smooth undulation on a 1m x 1m surface model poses a difficulty in reconstruction for most surface enhancement algorithms. However, the automated kriging method can model the underlying surface quite well. For the experiments involving this model, the scanning simulator parameters were set to be: a vertical resolution of 5 mm, a horizontal resolution of 2 cm, and a range/orientation uncertainty of 0.5%. This type of non-uniform sampling is typical of many laser scanning systems, and is the cause of the majority of the undersampling problems mentioned previously. The output of the scanning simulator can be seen labeled as "Scanned Geometry" in Figure 5.5.

The geometry enhancement methods chosen to compare the automated kriging results with are Linear (triangulation-based linear) and IDW interpolation. These two methods are

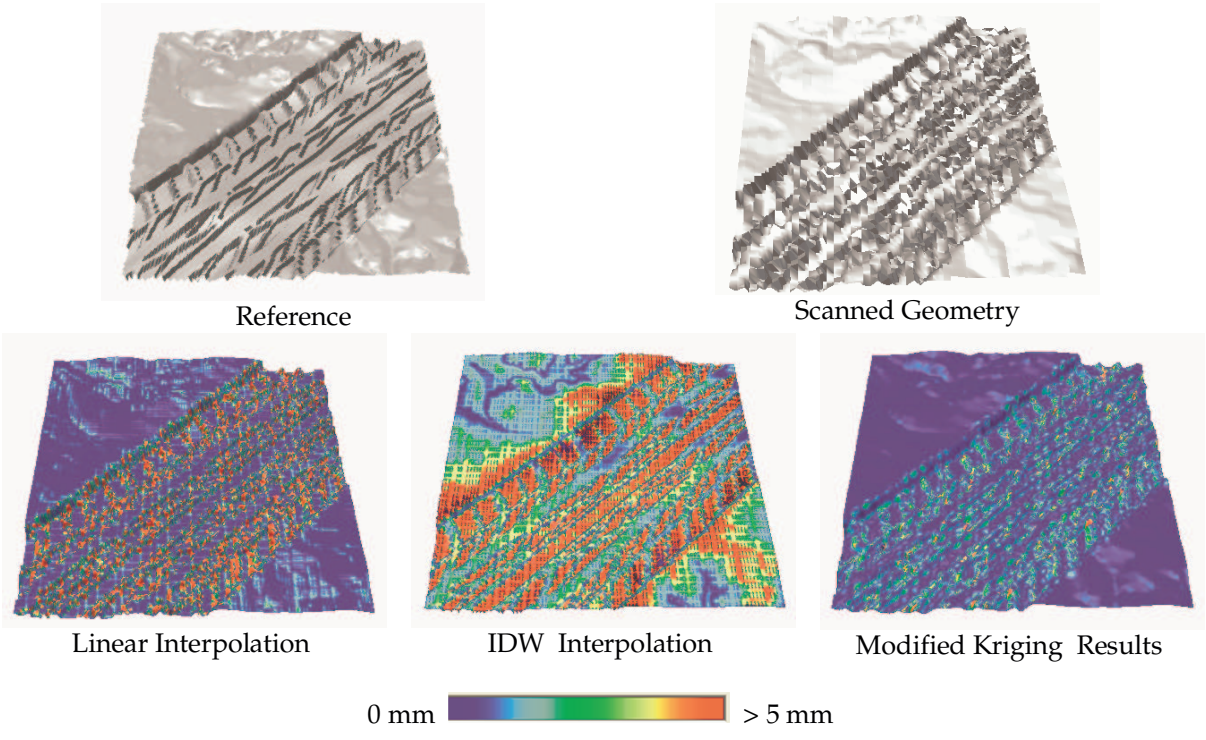


Figure 5.5. Results of geometry refinement methods applied to a synthetic test dataset. A synthetic reference model was scanned using a simulator that replicates the effects of a mobile 3D scanning system, and the output was used as the data for a selection of geometry refinement algorithms. Variations from the reference model are color coded.

standard methods for surface resolution enhancement and thus make good baselines to test against. Each of the methods were trained on the entire dataset before the estimation of the grid points. Each of the three interpolants was used to generate a 3D surface model with a sampling density of 5 mm in each direction. The color variations in the surfaces denote the deviation from the reference model in mm, with blue denoting low deviation and red identifying deviations of 5 mm or greater.

The results in Figure 5.5 show that the modified kriging algorithm does a better job of surface resolution enhancement overall, with fewer areas of red and more of blue, as compared to Linear and IDW interpolation. This supports our claim that the automated kriging method provides superior performance in geometry enhancement, yielding a 3D surface that more closely represents the underlying model. Note the characteristic "bulls eye" effect of the IDW reconstruction, whose poor performance can be attributed to the sparse and non-uniform nature of the sampling.

As a quantitative comparison, Table 5.1 shows the tabulated results of the deviations of the reconstructed surfaces to the original reference model. Notice that the automated kriging method has lower error statistics than either Linear or IDW interpolation, with a median improvement of 2% or more in accuracy. These results show that the automated kriging method outperforms the other interpolants quantitatively, as well as qualitatively.

Having shown that the modified kriging method can outperform the standard methods, at least in the general sense, we can now demonstrate the effects of the training sets on the resolution enhancement process. Six training regions, seen in Figure 5.6, were arbitrarily selected with three each chosen from the smooth (Smooth1 – Smooth3) and detailed surfaces (Detail1-Detail3) of the Tire Tread model. The automated kriging method was trained on each of these regions independently and the model was reconstructed at a sampling density of 5 mm.

The leftmost graph of Figure 5.6 shows the variograms generated from the six training sets. There is a clear similarity between the variograms calculated from the region types. The variograms from Smooth1-Smooth3 exhibit similar performance, smoothly varying from 0 to the sill. Likewise, the Detail1-Detail3 variograms show similar behaviors, rising quickly

Table 5.1. Quantitative comparison of interpolation results for the TireTread experiments. Statistics shown are the Minimum, Maximum, Mean, and Median errors along with the Standard Deviation (SD), Coefficient of Variation (CV), and RMS of the errors.

Method	Min (mm)	Max (mm)	Mean (mm)	Median (mm)	SD	CV	RMS
Linear	9.59 e-5	25.95	2.52	1.52	0.10	0.040	4.04
IDW	13.42 e-3	44.83	9.84	8.19	0.24	0.024	12.43
Modified Kriging	6.19 e-4	25.18	2.50	1.17	0.08	0.031	3.78

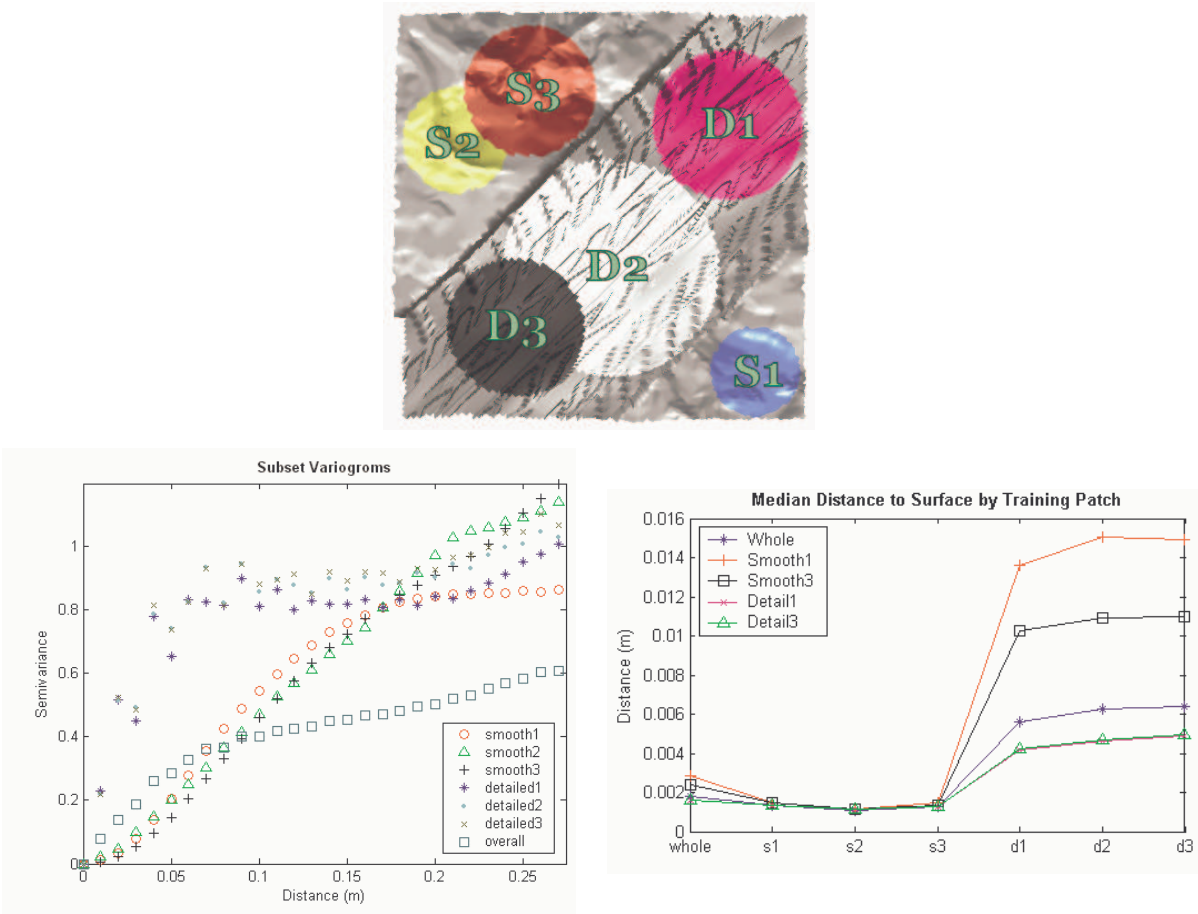


Figure 5.6. Six regions chosen to test the effect of training regions on the spatial enhancement, their respective estimated variograms, and the resultant error after the resolution has been enhanced.

before oscillating about the sill. Also shown in the graph is the variogram calculated from the whole dataset. The "overall" variogram exhibits a smoothly varying structure, similar to that constructed from the "smooth" regions. Thus, in order to preserve the features in the regions of the most interest – the "detail" regions in this case – the underlying model would be best represented by a variogram constructed only from data lying within those regions.

The right graph in Figure 5.6 relates the median distance of the surfaces reconstructed from the individual training patches to the original model. One training set from each category was removed to clarify the presentation of the results. The graph shows that using training data only from within the detailed regions (the ROIs) yields superior performance in the areas of the model containing the detailed surfaces and similar results in the smooth regions when compared to the reconstructions from the other training sets. This evidence supports

the claim that restricting the training data only to the specified ROIs is beneficial to the resolution enhancement process.

5.3.2 Experiments Using Real Data

The next set of experiments involves downsampling an existing 3D model in order to determine the effects of a reduced sampling on a less symmetric dataset. The premise is to investigate the ability of the automated kriging algorithm to preserve the detail and structure of real world data under lower resolution conditions. The experiments for this section involve taking high-resolution scan data from real applications and reducing the resolution of the model by only utilizing a fraction of the original data.

Figure 5.7 shows a 400 mm x 300 mm model of a subset of a stretch of pavement, imaged at

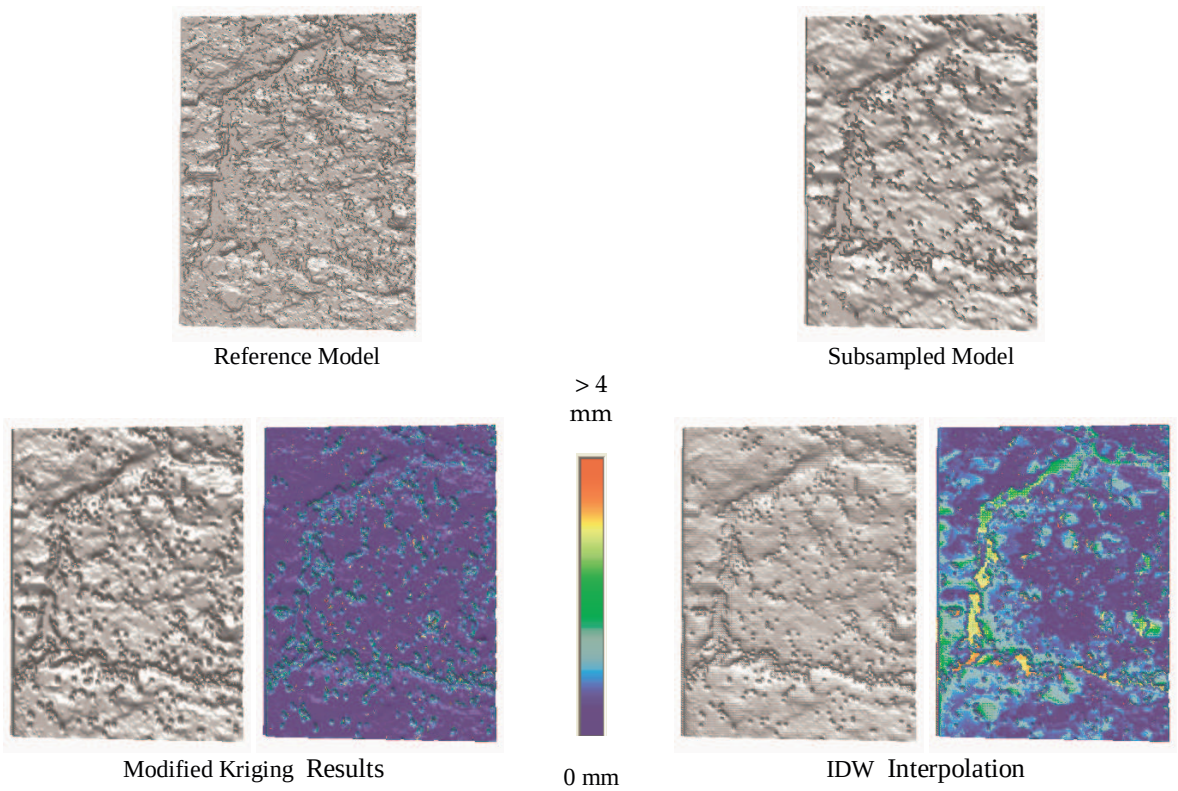
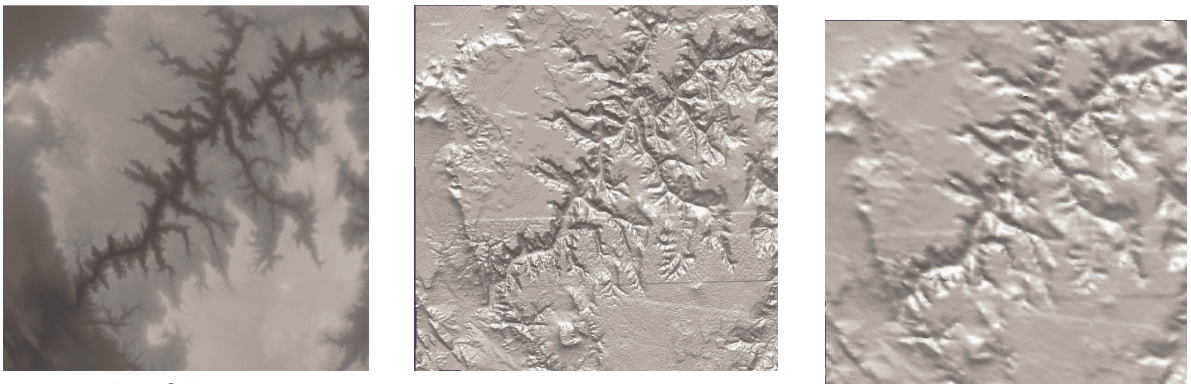


Figure 5.7. Comparison of geometry enhancement methods on a resampled dataset. A high-resolution model of a patch of pavement is resampled at 1/3 the original sampling and then geometry enhancement via our automated kriging and IDW are compared. Variations from the reference model are shown color coded to the right of each reconstructed model.

a 1 mm resolution in the horizontal and vertical directions over several meters. This high-resolution model shows a large crack that runs in a loose arc through the model, with an average depth of 14 mm. The original data was resampled at intervals of 3 mm and then geometry enhancement was performed on the resampled data. This resampled data could have been obtained by scanning 3 times as fast or with a scanner with a third of the resolution of the experimental system. Figure 5.7 shows the reconstructed surfaces obtained from our method and from IDW as well as a color coded surface showing the deviation from the reconstructed model to the original data. Our automated kriging method provides a close reconstruction as seen by the color-coded error map. In comparison, the IDW reconstruction has more error and fails to adequately represent the small-scale details present in the reference model.

Additional study models were examined for the use of our geometry enahncement method in a number of areas. One of these is Digital Elevation Map (DEM) building. Figure 5.8 shows a sample DEM model chosen for automated geometry enhancement. Surface models of this type are commonly generated for surface inspection, geometric visualization, bump map building, etc.

The Grand Canyon DEM was obtained from the United States Geological Survey (USGS), processed by Chad McCabe of the Microsoft Geography Product Unit, and used in several 3D processing publications (H.Hoppe, October 1998). The entire dataset encompasses nearly 245 km x 122 km sampled at 60 m intervals, but for this example we used a subset of the data approximately 60 km per side, totaling 1 million samples. The height resolution of this dataset is approximately 10 m with a height range of 1234 m.



Grand Canyon DEM Image | Original DEM Surface | DEM Sampled at every 6th pixel vertically and every 10th pixel horizontally

Figure 5.8. 3D model of the Grand Canyon DEM dataset and the lower resolution samplings used for geometry enhancement. The Grand Canyon DEM was originally sampled at 60 m between measurements.

The test model was sampled as if from a mobile scanning system, with a high number of samples in the scanning profile and a varying degree of separation between the sampling profiles. The surfaces were then enhanced using our algorithm and the benchmark interpolants, and the resulting surfaces were compared to the original data.

The Grand Canyon model was sampled at a resolution of 360 m per sample along the profile (every 6^{th} pixel) and 600 m between profiles (every 10^{th} pixel) and the resulting surface can bee seen in the right image of Figure 5.8. The resulting 15,000 sampled locations were then used to reconstruct the surface at its original resolution of 60m. Figure 5.9 shows the resulting reconstruction along with a qualitative comparison of the enhancement methods. The qualitative comparisons are given in Table 5.2, demonstrating that the automated kriging method outperforms the benchmark methods in terms of accuracy. Our method yielded a more accurate reconstruction, with less deviation from the original measurements in all categories. In addition, the comparison data shows that the median error for our modified kriging reconstruction is actually less than the original resolution of the 3D model, 8.9 m compared to the original vertical resolution of 10 m, showing that on average our reconstruction is comparable to the original measurements while only utilizing $1/600^{th}$ of the data.

For all of our test cases, our automated geometry enhancement algorithm performed better in terms of reconstruction accuracy. The numbers for the SD and CV show that the distribution of the errors is tighter for our algorithm than the benchmark methods. This means that the distribution of errors is more tightly centered, indicating a better overall fit. In addition to the numbers, the visual comparisons of the recovered surfaces show that our method reconstructs the surface closer to the original than the other interpolation methods. The fine details of the original surfaces are preserved using our method, but are obscured or not present at all for the Linear and IDW reconstructions.

Table 5.2. Quantitative comparison of the automated kriging geometry enhancement with standard interpolation methods for the Grand Canyon model. Statistics shown are the Minimum, Maximum, Mean, and Median errors along with the Standard Deviation (SD), Coefficient of Variation (CV), and RMS of the errors.

Model	Method	Min (m)	Max (m)	Mean (m)	Median (m)	SD	CV	RMS
Grand Canyon	Linear	0	453.0061	21.1990	10.8767	29.9373	1.5299	36.6830
	IDW	0	462.6975	23.4866	13.2276	33.1625	1.4120	40.6371
	Modified Kriging	0	370.7250	16.5717	8.9313 m	22.8731	1.3802	28.2454

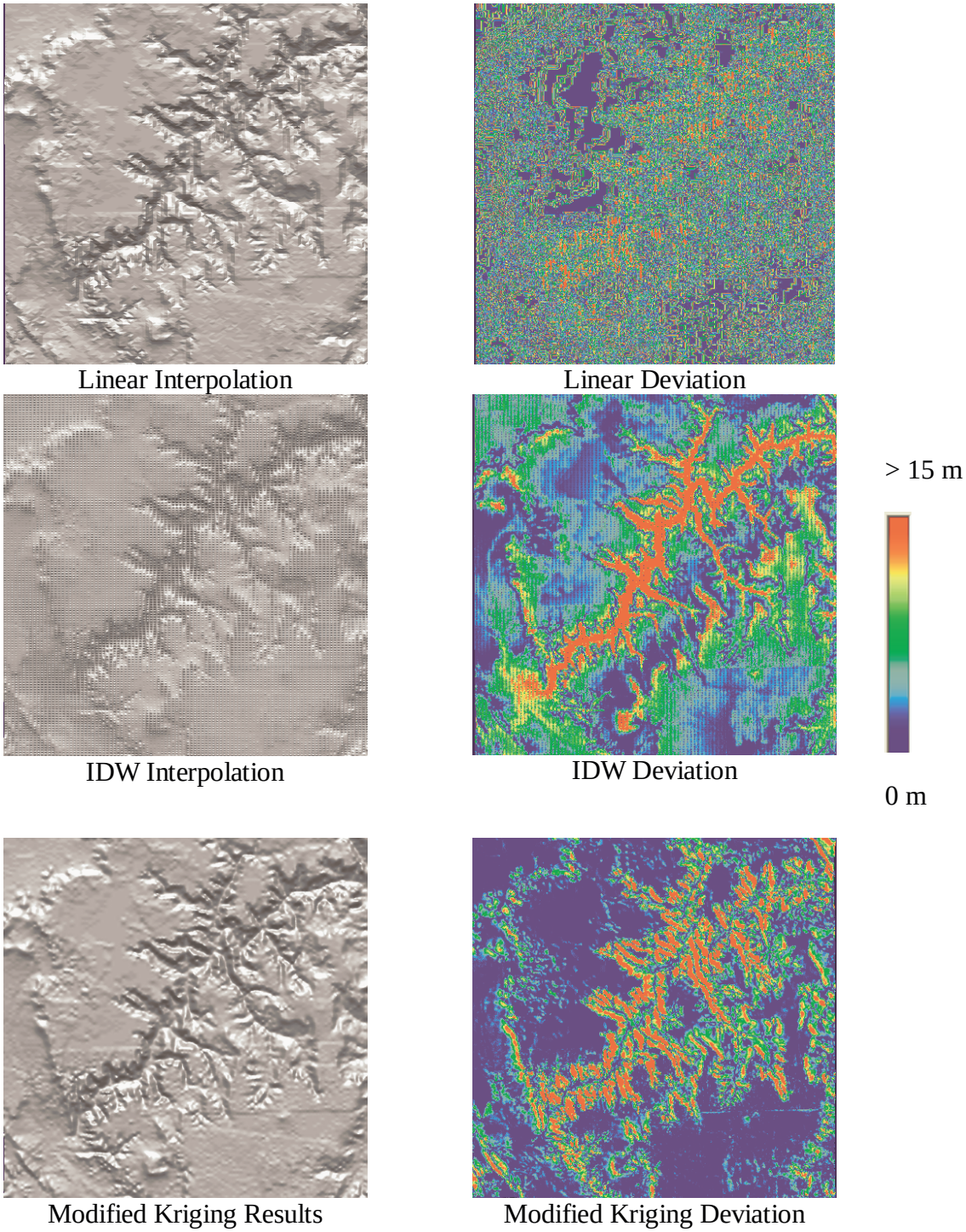


Figure 5.9. 60 m x 60 m Surface reconstructions and color coded deviations for the Grand Canyon model in Figure 5.8.

5.3.3 Effect of Sampling Resolution on Real Data

One of the most intriguing features about our automated geometry enhancement method is its ability to preserve details that are at the edge of the lowest scanning resolution of the measurement system. For many applications it would be useful to know in advance what scale of details could be recovered through geometry enhancement, given a certain sampling resolution. This would allow experiments to be set up to take advantage of the speedup possible when lowering the sampling resolution – increasing the traversal speed of the system – of the system. To this end, we have performed some tests to determine the smallest feature size that can be recovered at specified levels of scanning resolution.

The procedure for the Feature Size versus Sampling Resolution (FSSR) experiments is as follows: (1) Take an existing 3D dataset with a large variation in feature sizes and manually identify zones containing features of a specified size; (2) Resample the model; (3) Perform our automated kriging geometry enhancement to recover a surface of the original sampling density; (4) Visually compare the recovered surface to the original model, noting the affects on the previously identified zones of features. In order to provide a common and robust terminology for the FSSR experiments, we will refer to the smallest sampling resolution of the system to be a pixel. In fact, for the Grand Canyon DEM used in the experiments, the data was given as an image, making this an intuitive choice.

The first set of FSSR experiments involved isotropic sampling of the DEM. Pixels were sampled at regular intervals in multiples of 2 up to 24, as seen in Figure 5.10. The number of pixels at the highest resolution is 512x512, while the sampling at the lowest resolution yielded a dataset of on 42x42 pixels. Table 5.3 shows the results of the experiments, with the level at which each feature size appeared significantly smoothed, severely degraded, and then disappeared entirely. A value of '-' indicates that there was no significant loss of features at that sampling level. Table 5.3 also shows the overall statistics for the recovered surfaces, as compared to the original. This gives an idea of not only the feature sizes present at each sampling level, but the relative overall performance of the reconstruction. A visual representation of Table 5.3 is seen in Figure 5.11, where the maximum scanning resolution is shown for each size feature, for all 3 states of feature recovery - smoothed, degraded, and disappeared (missing).

The isotropically resampled data became smooth very quickly, with most of the linear and small surface features becoming smoothed out in the first 3-5 sampling intervals. It should be noted that some of the structures at a given feature size remained present in the recovered surface long after other features of similar size disappeared. This is most likely due to a "conveniently" sampled point that happened to fall on the feature, with the rest of the feature being recovered through the kriging's signal inference properties. Also note that the roughly horizontal lines in the lower third of the image corresponding to swatches of altimetry data being slightly misaligned effectively disappeared at the 10x10 sampling interval.

The next FSSR experiment was done using anisotropic samples of the DEM. Every other pixel was sampled in the vertical (scanline) direction, while interprofile spacing was set up in intervals of multiples of 2 up to 24 as seen in Figure 5.12. The number of pixels at the

highest resolution was 512x512 – which is the same as that from the previous experiment – and the dataset at the lowest resolution was 512x42. The FSSR results for the anisotropically sampled case are given in Table 5.4, as well as the overall statistics of the deviation of the recovered surfaces from the original. A visual representation of Table 5.4 is seen in Figure 5.13, where the maximum scanning resolution is shown for each size feature, for all 3 states of feature recovery - smoothed, degraded, and disappeared (missing).

For the anisotropic sampling, the features were preserved longer than in the regularly sampled case. The main reason for this is that the density of samples along profile partially made up for the loss in sampling density between profiles. This is most evident in the previously mentioned horizontal lines from scan mismatches. These lines are clearly present up through the 2x18 sampling case, and still evident even in the 2x24 sampling. The increasing distance between scanlines could not overwhelm the densely sampled profiles for these horizontal structures. Vertical structures, such as some of the chasms, did not fare so well as the interprofile spacing increased, leading to strong "banded" canyons in the recovered surfaces, rather than the diagonal or fluted structures present in the original data.

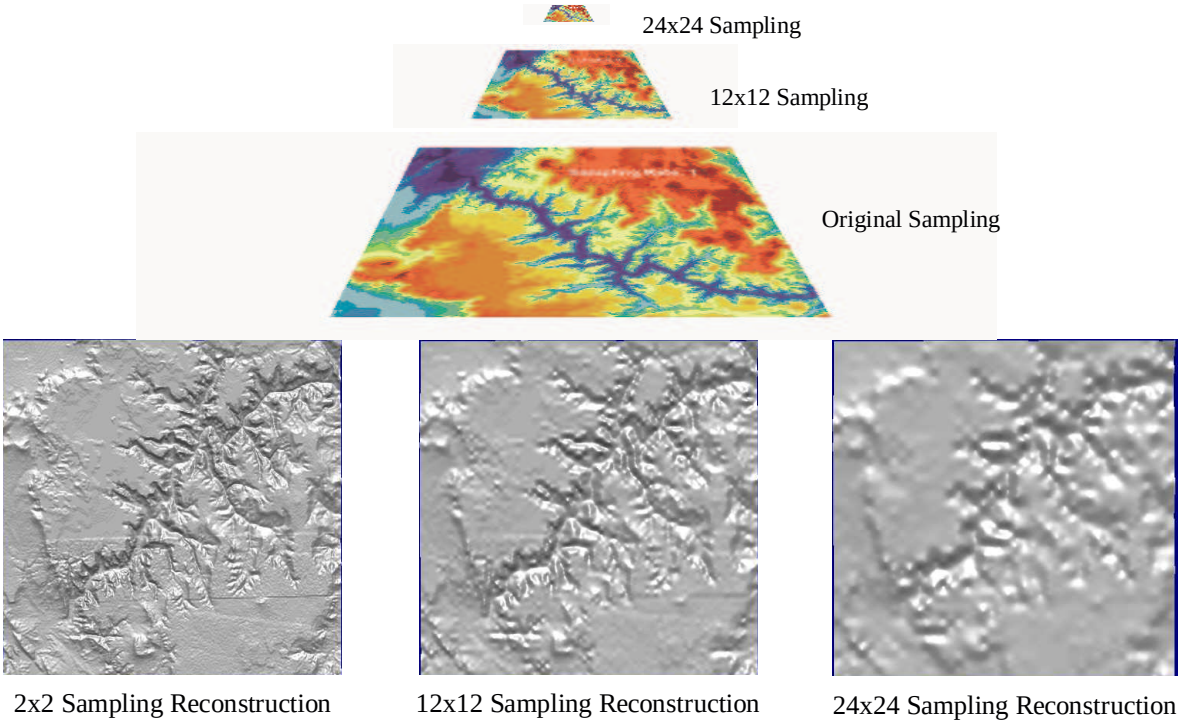


Figure 5.10. Recovery of features from isotropic sampling of the Grand Canyon DEM. The DEM was sampled at regular intervals in multiples of 2, up to 24. The original number of samples was 1024x1024, while the number of samples at the lowest resolution was 42x42.

Table 5.3. Recovery of features from isotropic sampling on the Grand Canyon dataset and our geometry enahncement method. Values given show the sampling density at which features are smoothed, degraded, and disappear. Feature size given in pixels. Also given are some statistics of the global error of the enhanced models, including the Median error, Standard Deviation (SD), Coefficient of Variation (CV), and RMS of the errors. A (-) entry indicates no observations at that pixel level.

Sampling Resolution	Smoothed (pixel level)	Severely Degraded (pixel level)	Disappeared (pixel level)	Median (m)	SD (m)	CV	RMS
2 x 2	1	-	-	2.2028	5.2916	1.6522	6.1854
4 x 4	2	1	-	4.2108	8.0461	1.2348	10.3536
6 x 6	4-5	3	1-2	6.0288	12.8832	1.2956	16.2744
8 x 8	6-7	4-5	3-4	8.0838	19.1770	1.3366	23.9500
10 x 10	8-9	6-7	5	10.1943	25.9823	1.3521	32.3163
12 x 12	10-11	8-9	6-7	13.1286	34.7117	1.3501	43.1968
14 x 14	12-13	10-11	8-9	14.5218	38.5483	1.3338	48.1797
16 x 16	13-15	10-12	12-13	16.5796	45.2354	1.3430	56.3976
18 x 18	16-17	14-15	10-13	19.0809	52.7509	1.3455	65.7254
20 x 20	18-19	16-17	14-15	21.2523	57.6190	1.3303	72.0836
22 x 22	20-21	18-19	16-17	23.5972	66.1271	1.3499	82.2948
24 x 24	21-23	19-20	18	25.5447	73.1070	1.3604	90.7326

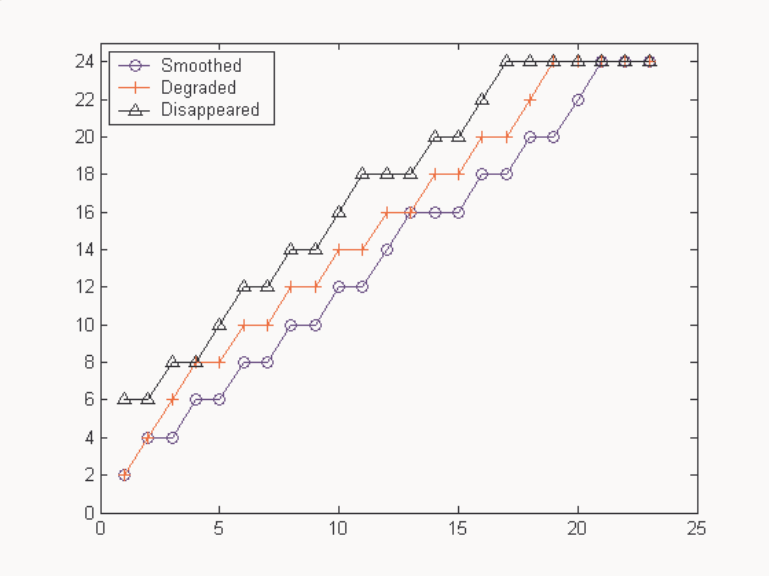


Figure 5.11. Maximum isotropic sampling resolutions at which features in the dataset shown in Figure 5.10 and Table 5.3 were smoothed, degraded, or disappeared.

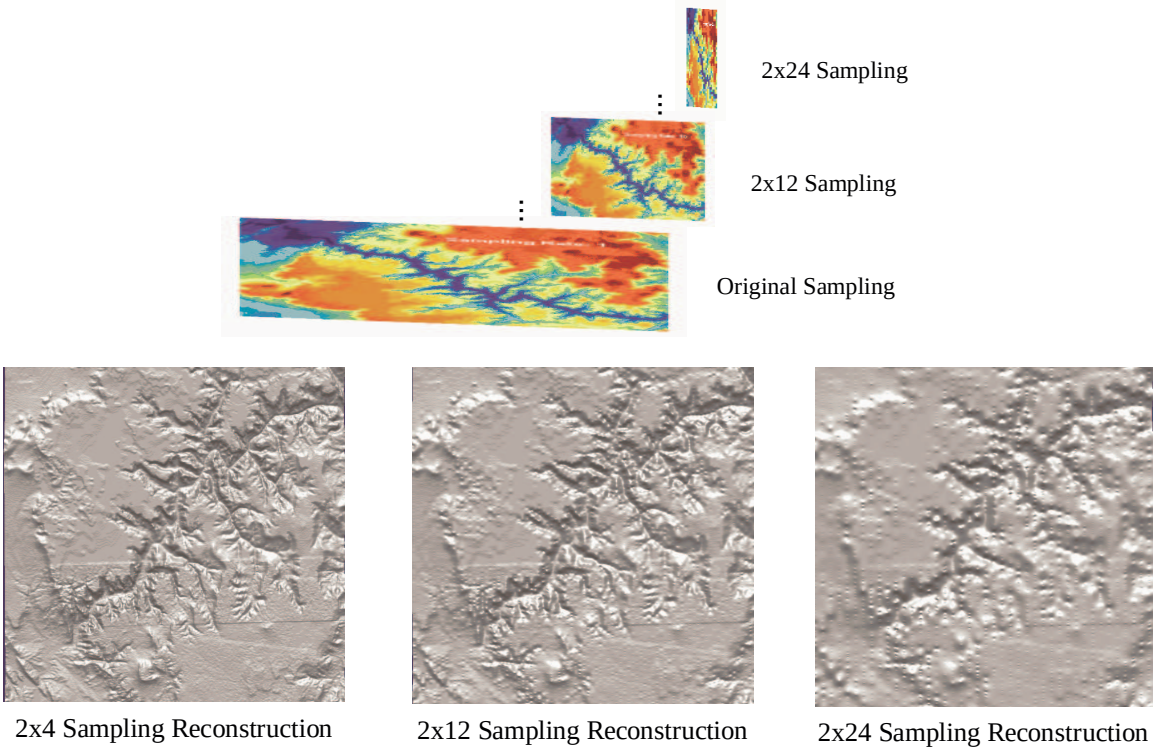


Figure 5.12. Recovery of features from anisotropic sampling of the Grand Canyon DEM. The DEM was sampled at regular intervals in multiples of 2, up to 24 in the horizontal direction, while sampling every other value in the vertical.

Table 5.4. Recovery of features from anisotropic sampling on the Grand Canyon dataset and our resolution enhancement method. Values given show the sampling density at which features are smoothed, degraded, and disappear. Feature size given in pixels. Also given are some statistics of the global error of the enahnced models, including the Median error, Standard Deviation (SD), Coefficient of Variation (CV), and RMS of the errors. A (-) entry indicates no observations at that pixel level.

Sampling Density	Smoothed (pixel level)	Severely Degraded (pixel level)	Disappeared (pixel level)	Median (m)	SD (m)	CV	RMS
2 x 4	1-2	-	-	3.7650	8.7152	1.4806	10.5168
2 x 6	3-4	1-2	-	4.5891	10.8642	1.3929	13.3740
2 x 8	5-6	3-4	1-2	5.8848	15.5625	1.4498	18.9054
2 x 10	6-8	3-5	-	7.2189	21.0453	1.4867	25.3633
2 x 12	7-10	5-6	3-4	8.4607	26.7983	1.5028	32.1890
2 x 14	10-12	7-9	5-6	9.8485	32.1962	1.4975	38.7151
2 x 16	12-14	9-11	7-8	11.2748	37.4331	1.4972	45.0151
2 x 18	13-16	11-12	9-10	12.7415	43.6629	1.5033	52.4405
2 x 20	17-18	14-16	11-13	13.8475	47.8375	1.4930	57.5769
2 x 22	19-20	17-18	14-16	15.1812	53.7285	1.5071	64.4798
2 x 24	22	20-21	17-19	16.9878	62.6722	1.5428	74.6862

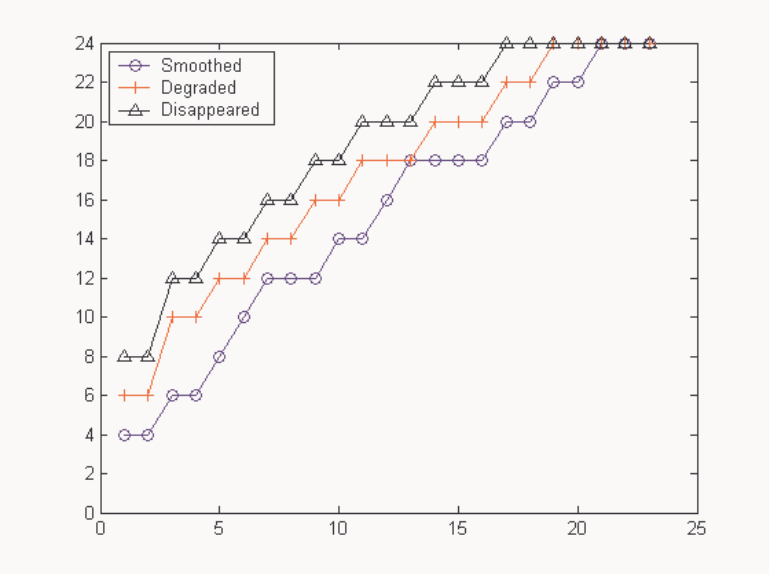


Figure 5.13. Maximum anisotropic sampling resolutions at which features in the dataset shown in Figure 5.12 and Table 5.4 were smoothed, degraded, or disappeared.

The FSSR experiments give an idea of the size and relative condition of features that can be recovered at a given sampling interval. It should be stated that these results only hold for semi-continuous features. This is not a magic result that allows small features to be preserved without any sampling whatsoever. Isolated features of a given resolution that fall between scanlines will not likely be recovered. However, if at least 1 sample can be placed on or near a structure at a given resolution, the method has shown that it can recover structures, whose dimensions are smaller than the sampling interval, albeit at varying degrees of accuracy.

5.3.4 Discussion on Computation

In each of the experimental results given in this paper, our automated kriging method outperformed all other competing methods in terms of geometrical accuracy. However, for the datasets shown here the data was sparsely sampled in an irregular fashion. Early experiments showed that when the sampling density is sufficiently high, there is minimal difference in the accuracies between the various interpolation algorithms. This observation should be intuitive, as all interpolation algorithms perform best when the input sample density is close to the output sample density (Zimmerman et al., 1999). Thus, we expect our method to be most useful when the need for geometric accuracy is high, the sampling density is relatively low, and the computation time is not a restriction.

At this point, it should be stated that our geometry enhancement method is computationally intensive, forcing a trade-off between accuracy of reconstruction and time performance. The decision on whether the extra computation is an acceptable cost for the improvements in the geometry enhancement will need to be weighed carefully, balancing the cost/accuracy ratio on the needs of the specific application. To compare computational needs for the geometry enhancements we have used in this paper, let us consider a set of data consisting of n samples. We use these datum to evaluate the surface at m new locations (typically $m > n$). The order of complexity is analyzed in 2 parts: the complexity of the training stage, and the complexity of the evaluation of new estimates.

Triangle based linear and cubic interpolation requires the Delaunay triangulation of the dataset – an $O(n^2)$ operation that can be optimized for $O(n \log n)$ time (Franke & Nielson, 1980) – followed by searching the triangulation space for the closest triangle and solving for the polynomial interpolator of the specified degree for a total of $O(n \log n + m)$. IDW essentially "trains" the weights for every new estimate, for a total of $O(nm)$ operations (Shepard, 1968). Kriging requires the training of the variogram – an $O(n^2)$ process – followed by the solution of a large, sparse linear system for every new estimate – an $O(m^3)$ process. Neighborhood restriction and optimized linear system solvers can reduce this to a total of $O(n^2 + m \log m)$ operations (Tzeng et al., 2005).

Figure 5.14 shows a timing comparison, done in Matlab on a 2.4 GHz Pentium IV with 1 GB of RAM, for the interpolators used in this paper. The evaluation shown is the time taken for each method to train on the available samples and evaluate 1 new point. The number of samples ranged from 10^1 to 10^5. While the variance in the code optimizations for the

different methods makes the actual numbers difficult to compare objectively, the graph does show the fact that IDW is the fastest method, followed by triangle-based interpolation and then by kriging, with a timing separation of nearly an order of magnitude between each class. This, of course, is simply the training overhead. As the number of points to estimate grows large, triangulation-based interpolants take the lead in processing time, with kriging and then IDW following after.

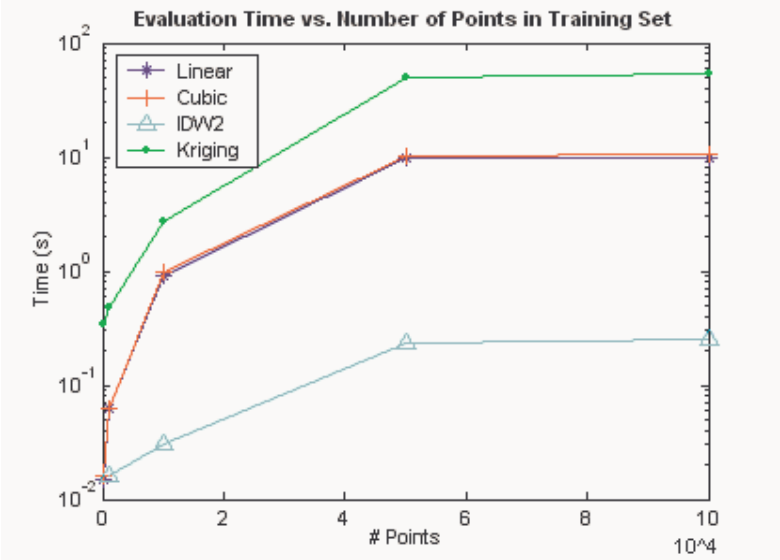


Figure 5.14. Comparison of training time for the interpolants.

6 Extension of Empirical Mode Decomposition to 3D Surfaces

6.1 Empirical Mode Decomposition

In research and many practical applications data analysis is an important and necessary part. Data acquired from real world experimentation often appears chaotic and noisy, yet it represents to us a semblance of the information we are trying to obtain. Analysis of these data is required to identify the parameters needed to construct a specific model of the system we are interested in.

Since its introduction, Fourier spectral analysis has dominated signal analysis efforts. In fact, the term *spectral analysis* has become synonymous with Fourier analysis. Unfortunately, there are some restrictions to Fourier analysis that can make it unsuitable for examining the results of real-world experimental data: the signal must be linear, and the data must be stationary. These non-linear, non-stationary signals require a more elaborate and meaningful method for examining their frequency spectra than Fourier analysis can provide.

In 1998, Huang et al. (Huang, et al., 1998) introduced a method for adaptively decomposing a signal into well-behaved zero-mean AM-FM components (details) riding on a low frequency bias (trend). Huang termed this method Empirical Mode Decomposition, and its output can be analyzed using a Hilbert-Huang Transform to determine the instantaneous frequency spectrum of the signal at any point. EMD is a data-driven signal decomposition. Unlike Wavelet decomposition, which uses arbitrarily chosen basis functions to perform the decomposition, EMD's basis functions are drawn directly from the data, making it a highly adaptive signal decomposition algorithm.

The foundation of EMD is to consider signal oscillations at a local level. For instance, if we look at a section of the signal contained between two minima, we can define the high-frequency part (detail) $d(t)$ as the oscillation terminating at the two minima points, and passing through the maxima point that lies between them. Underlying this high-frequency component is the trend $m(t)$, which can be seen as $x(t) = m(t) + d(t)$. Assuming that this can be done for all oscillations contained in the signal, the procedure can then be iteratively

applied to the residual until there are no more oscillations present. Thus, EMD iteratively identifies the intrinsic oscillatory modes present in the data in an empirical fashion, and decomposes the data accordingly.

These intrinsic oscillatory modes are known as Intrinsic Mode Functions (IMFs), which are signals that satisfy: (1) in the whole data set, the number of extrema and the number of zero crossings must be either equal or differ at most by one; (2) at any point in the data set, the local mean – defined as the average between the envelope defined by the maxima and that defined by the minima – is zero. The IMF is an amplitude- and frequency-modulated signal that is (likely) non-stationary.

To generate an IMF from a signal $X(t)$, first the extrema are identified, and the maxima and minima envelopes – e_{max} and e_{min}, respectively – are generated. Their mean is designated as m_1, and the difference between the data and m_1 is the first iterate

$$h_1 = X(t) - m_1. \tag{6.1}$$

Figure 6.1 shows the procedure to find h_1, with the original signal shown in Figure 6.1a, the envelope generation in Figure 6.1b, and the difference between the data and m_1 in Figure 6.1c.

In the ideal case, h_1 would be an IMF, however, overshoots and undershoots are common, and it is evident from Figure 6.1c that there are still positive minima and negative maxima, suggesting riding waves in the data. Iterating the process on the residual can isolate the riding waves from the iterate and make the signal symmetric about zero. The process is known as sifting, and occurs one or more times until the iterate is an IMF (shown in Figure 6.1d).

The algorithm for generating a complete Empirical Mode Decomposition from a signal $X(t)$ can be summarized as:

- while (!done)
  1. find extrema of $R(t)$
  2. generate extrema envelopes e_{min} and e_{max}
  3. find local mean $m_i = (e_{min} + e_{max})/2$
  4. extract the detail $h_i = X(t) - m_i$
  - if ($\Delta h <$ threshold) //little change in iterates
    - $c_k = h_i$
    - $X(t) = X(t) - c_k$
    - Repeat steps 1-4 on new $X(t)$
  - else
    - Repeat steps 1-4 on the residual h_i
  - end if
- end while

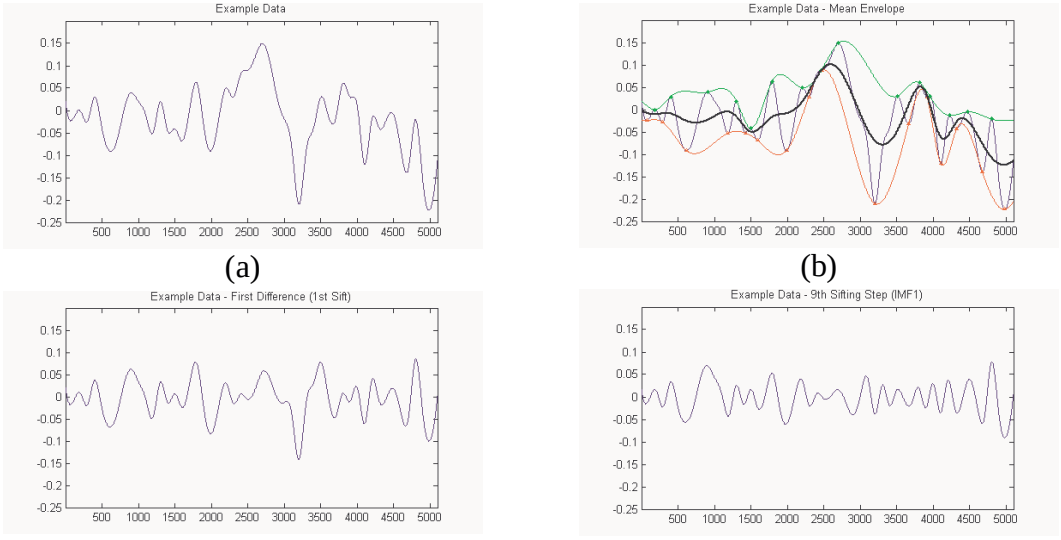


Figure 6.1. Illustration of the EMD sifting process on a synthetic signal: (a) the original data; (b) the data shown in blue, with the extrema and their envelopes shown in green and red (maxima and minima, respectively), and the signal mean shown as a thick black line; (c) the difference between the data and m_i. This is not an IMF because there are still negative local maxima, and positive local minima; (d) the final IMF after nine sifting operations.

To illustrate the EMD algorithm, we will use a set of wind speed data acquired from the National Oceanic and Atmospheric Administration (2004). The wind data can be seen in Figure 6.2. Clearly, the wind speed data are quite complicated, with many high-frequency components riding on top of a low frequency wave form. There are also no zero crossings in the data. The EMD decomposition for this data set is shown in Figure 6.3. The decomposition of this complex data set is only 9 IMF terms labeled C1-C9. Each IMF represents the instantaneous energy content of the signal with respect to a certain frequency range.

By definition, the completeness of the decomposition is guaranteed to the precision of the computer. As a check of completeness, see Figure 6.4. Here, we have reconstructed the signal by combining the IMFs from the lowest frequency bands to the highest. The original signal is shown in red, with the approximations given in blue. From the start, it can be seen that the trend approximates the overall signal very well. As the higher frequency components are added, the approximation becomes more and more like the original. At the fullest reconstruction level, the difference between the reconstructed signal and the original is on the order of 10^{-15}, the maximum level of precision of the computer.

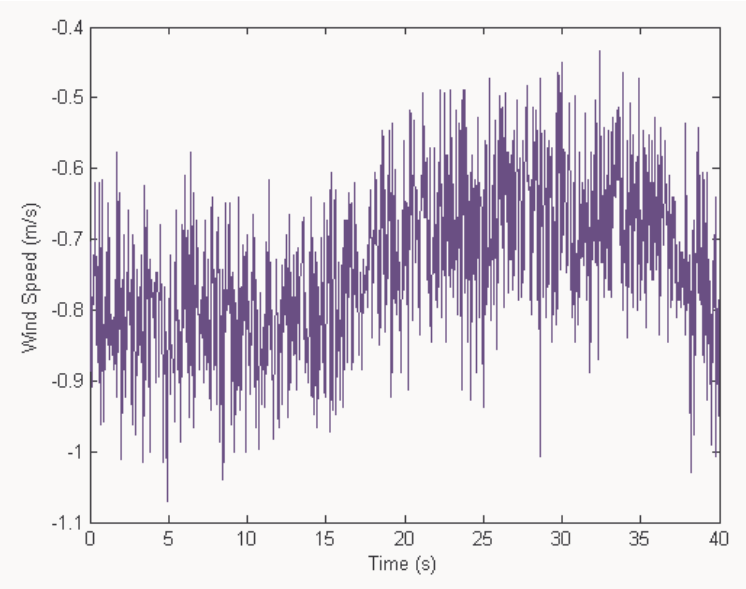


Figure 6.2. Wind speed data.

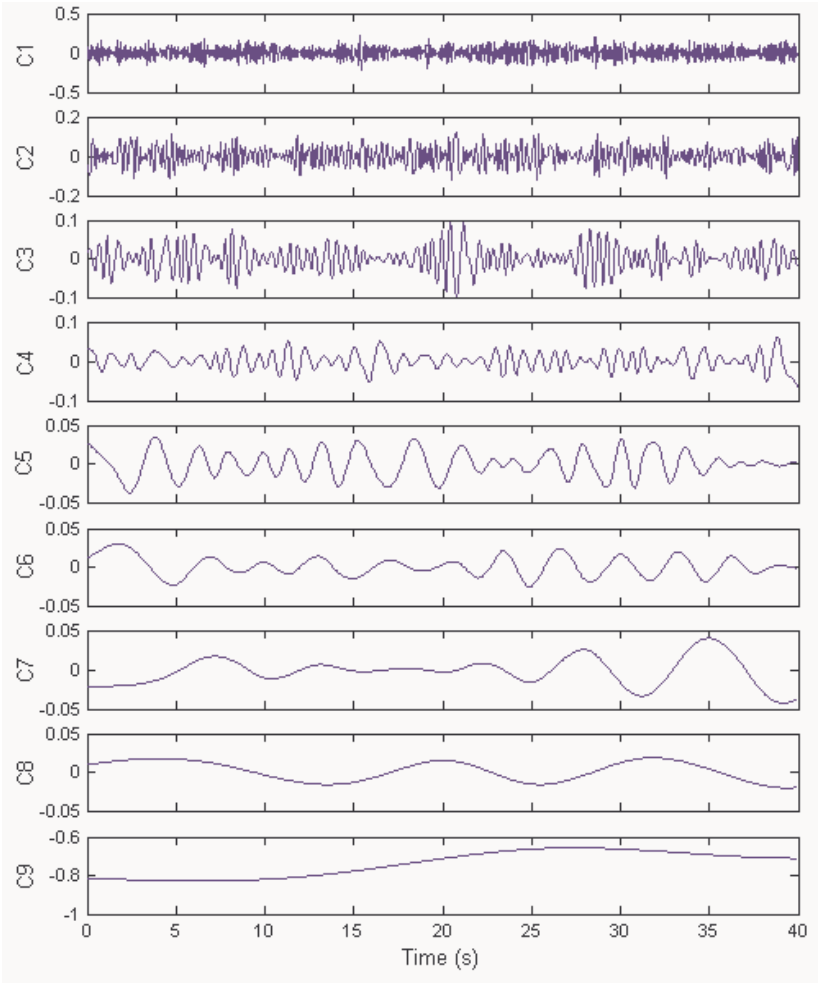


Figure 6.3. Empirical Mode Decomposition of the wind speed data.

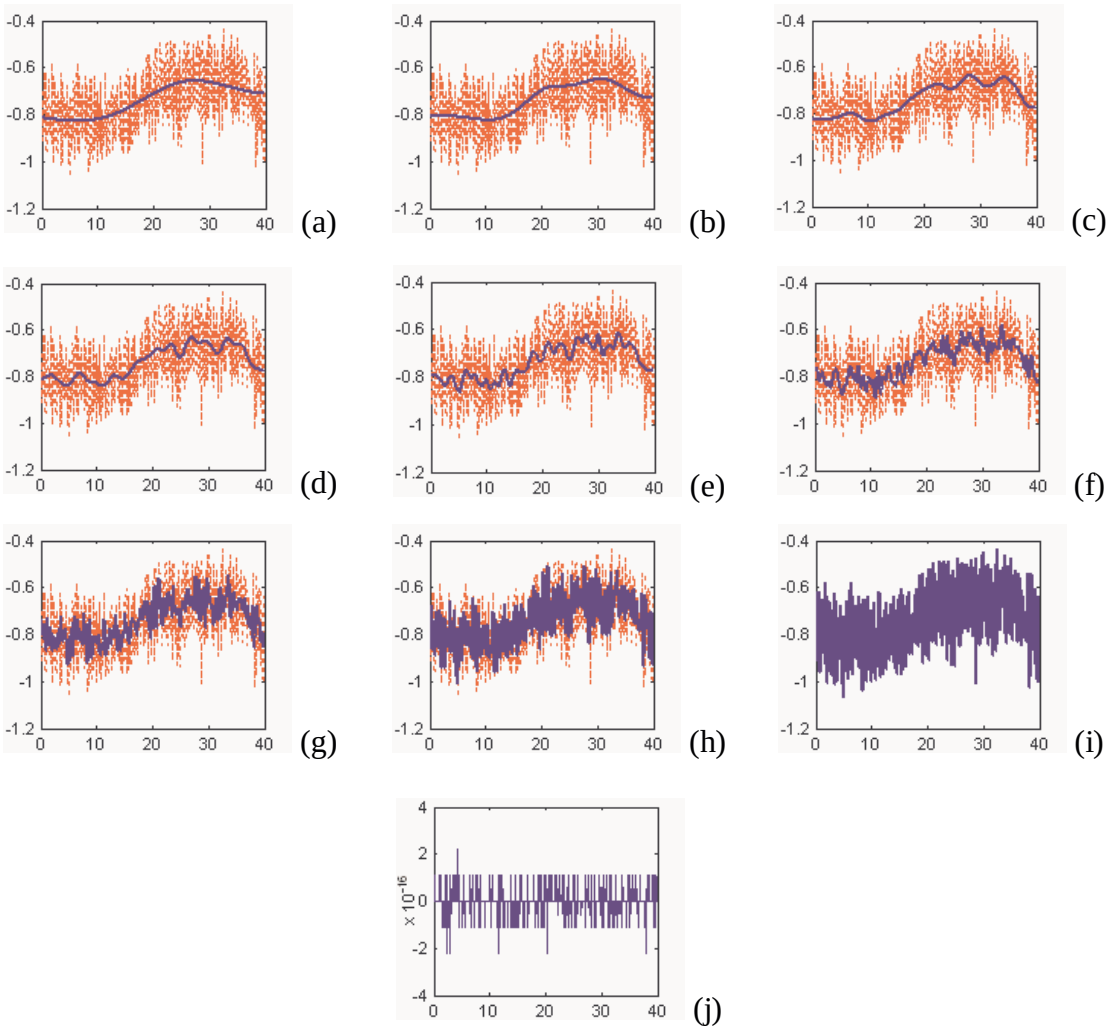


Figure 6.4. Numerical proof of the completeness of the EMD through reconstruction of the data from the IMF components. (a) Data (dotted red line) and the c_9 IMF component (solid blue line). (b) – (i) Data (dotted red line) and the signal reconstructed by the addition of 1 IMF per level of reconstruction (blue lines). (j) The difference between the original data and the reconstructed signal.

Empirical Mode Decomposition has been applied to many real-world signal analysis applications including: oceanic acoustics (Oonincx & Hermand, 2004), seismographic analysis (Huang, et al., 1998), cardiographic analysis and simulation (Neto et al., 2004), and solar activity (Coughlin & Tung, 2004). In addition, work has been done on evaluating the 1D EMD algorithm as a filter bank (Flandrin et al.), and on improving its performance (Rilling et al., 2003).

The extension of the standard 1D EMD theory to 2D grayscale images is a straightforward one, and has been investigated by several groups in the past few years. In essence, the image is treated as a 2D scalar field of intensity values, as seen in Figure 6.5. Nunes, et al (Nunes et al., 2003) used binary morphology to extract the maxima and minima of the intensity field, and then applied 2D radial basis functions for the surface interpolation. Another approach is to use simple comparisons within a 8-neighborhood to determine extrema, and then apply bi-dimensional cubic splines for the interpolation (Linderhed, 2002). These extensions of EMD have been applied for image processing tasks such as noise removal and image compression.

EMD has proven to be an effective data analysis tool for both 1D signals and 2D images. Unfortunately, no extension to 3D surfaces currently exists. There are existing methods for multi-resolution analysis of 3D surfaces (Guskov, et al., 1999; Roy, et al., 2003), but these methods do not provide the theoretical foundations of their 2D counterparts. The proposed extension of EMD to 3D surfaces will provide a rigorous foundation for surface decomposition, and the basis for energy-specific processing such as denoising, level-of-detail generation, etc.

Extending the Empirical Mode Decomposition algorithm to 3D surfaces is not as straightforward as the extensions to 2D previously discussed. There are several issues that are significantly different in the 3D case, and each must be addressed in order to perform the intended extension. These issues include:

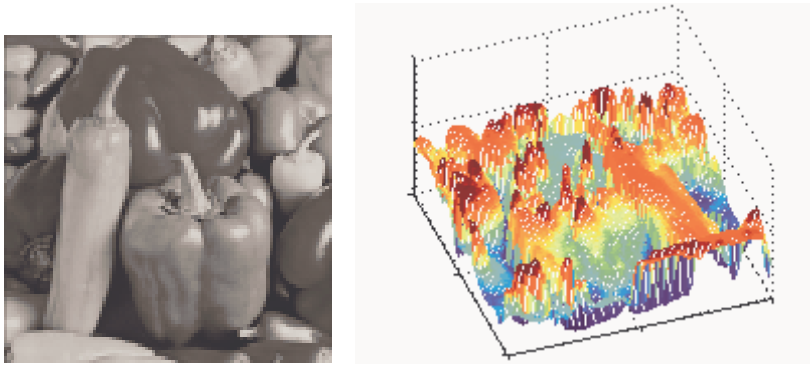


Figure 6.5. A grayscale image treated as a 2D height map.

- What are the fundamental differences between 1D and 2D EMD, and the extension to 3D EMD?
- How do you define extrema on a 3D surface?
- How do you connect extrema in a coherent fashion to form the envelopes?
- How do you define differences and means between extrema envelopes?

6.2 Mesh Feature Space

Empirical Mode Decomposition operates in a feature space, defined as the spatial locations of the signal domain combined with a set of attributes associated with the geometry. In the 1D case, such as that shown in the example of Figure 6.2, the spatial domain is a time series and the attribute associated with each time sample is wind speed. For the 2D example shown in Figure 6.5, the spatial domain are the image (x,y) indices and the analyzed attribute is the grayscale intensity at every index.

We define the feature space of a 3D surface as the 3D geometry - vertex locations in 3D space and the oriented surface patches (typically triangles) that link them together - and the set of attributes associated with each vertex. These attributes may be measured values such as: pressure, velocity, temperature, or radiance. Or, they may be values extracted from the 3D surface, such as: normals, curvature, centricity, or saliency. Figure 6.6 shows some examples of 3D surfaces with associated attributes. On the left is a mechanical part with associated surface temperature, in the center is a toy model displaying calculated centricity, and on the right is a model of a human with mean curvature being displayed on the surface.

A given mesh may have several such features assigned to it. Thus, the representation of a single vertex $\boldsymbol{v}$ of the mesh is $\boldsymbol{v} = (x_1, x_2, \ldots, x_m, f_1(v), f_2(v), \ldots, f_n(v))$. For the typical mesh analysis scenario covered by this research, only a single attribute function attached to a 3D surface will be considered. This reduces the general $n + m$ dimensional case to a 4D case, with the vertex now being represented as $\boldsymbol{v} = (x, y, z, f(v))$.

6.3 Defining Extrema on a 3D Surface

In both the 1D and 2D cases, the local extrema were extracted by simply evaluating their attribute values as compared to their neighbors', with the neighbors being defined as the closest points to the target vertex on a regular grid. In general, however, 3D surfaces are not constructed on a regular grid. Nor are attribute values always known for every vertex on the surface. Therefore, in order to extract the attribute extrema, it is necessary to define the neighborhood $\boldsymbol{N}$ around a surface vertex in regards to both geometry and attribute space.

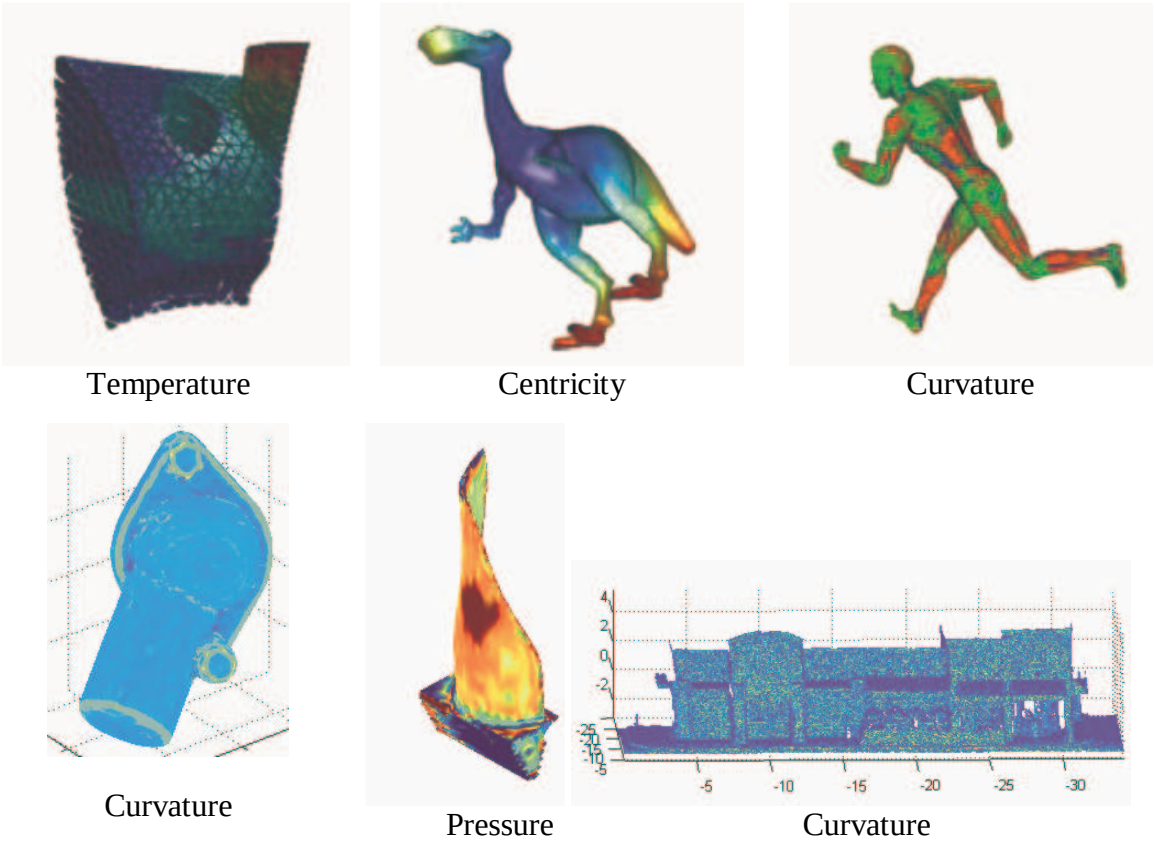


Figure 6.6. 3D surfaces with different attributes attached (top row adapted from (Shamir et al., 2006)).

6.3.1 Defining the neighbors

For a tessellated 3D mesh, the concept of neighborhood is readily understood as the set of points connected to a vertex by one or more edges. The 1-ring neighbors are those vertices connected to the target vertex by one, and only one, edge. In order to extract the extrema, a neighborhood mesh walking method is used to traverse the 3D surface and build up a list of neighbors for each vertex. Mesh walking is a procedure that has proved to be useful for a variety of 3D applications, including segmentation (Page, et al., 2003), clustering (Shamir, et al., 2006), and smoothing (Wang et al., 2003). In essence, mesh walking involves starting at a seed vertex, "walking" along the edges radiating out from that vertex to discover adjacent vertices, and building a map of connectivity information along the way. In the 3DEMD method, we utilize this procedure to identify vertex "neighbors" with associated attributes.

A 3D dataset need not be tessellated, however, to process neighborhoods. For a non-tessellated dataset (point cloud), the neighbors can be found through the definition of a

neighborhood interval. Let $\boldsymbol{v}$ be a vertex of point cloud $\boldsymbol{S}$ and $\boldsymbol{\delta}$ be the distance about $\boldsymbol{v}$ that neighbors are expected to lie in. Then, for every point $\boldsymbol{v_j}$ in $\boldsymbol{S}$, if $\| \boldsymbol{v_j} - \boldsymbol{v} \| < \boldsymbol{\delta}$ then $\boldsymbol{v_j} \in \boldsymbol{N_v}$, which is the set of neighbors for vertex $\boldsymbol{v}$. This idea of neighbors as "nearby" points does necessarily yield the same "connected by one edge" results as is available on a tessellated mesh, and the size f the neighborhood and density of neighbors is dependant on the parameter $\boldsymbol{\delta}$. However, if a tessellation is not available, or would be prohibitively costly to compute, the $\boldsymbol{\delta}$-ball approach to neighborhoods will suffice to set the stage for further analysis.

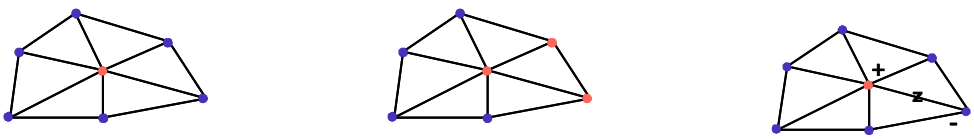
6.3.2 Isolating the extrema

Once a map of neighborhoods for each vertex on the mesh has been developed, the extrema can be isolated by comparing attribute values for each vertex with those of its neighbors. Figure 6.7 shows how extrema are detected on a 3D surface. If the attribute value of a given vertex is greater than that of all of its neighbors, it is identified as a local maxima. If the vertex attribute is equal to one or more of its neighbors but greater than the rest, it is part of a local maxima plateau (or ridgeline). Local minima are determined in a similar fashion. Zero crossings along an edge are determined by a sign change between the values of the edge endpoints.

When an extrema plateau is encountered, all vertices contributing to the plateau are retained for envelope generation, rather than isolating a plateau centerpoint or representative location. This provides more stability to the envelope generation and sifting processes, leading to a faster convergence of the sifting.

The simplest form of extrema detection would be to examine every vertex independently, compare it to all of its neighbors, and make a determination on whether or not it is an extrema. This will give a complete set of extrema, but at a large computational cost. For example, let us assume that a surface $\boldsymbol{S}$ has N vertices and every vertex has 6 neighbors. Independent examination of every vertex and it's neighbors would require 2*6*N comparisons - 6 comparisons for maxima evaluation and 6 for minima evaluation.

However, recognizing the fact that identifying one maxima automatically eliminates all of its neighbors as maxima candidates (excluding the possibility of plateaus for the moment), an



Local Maxima Maxima Plateau Zero Crossing

Figure 6.7. Extrema and zero crossing detection on a triangulated surface.

efficient algorithm can be developed that will eliminate a large percentage of the calculations. In fact, in the best case scenario 1/3 of the comparisons are eliminated. Experimentally, we have found that on average, this modified version of the extrema detection process eliminates roughly 20% of the calculations, as compared to the independent examination approach. Further computations can be removed, albeit a small percentage, by recognizing the fact that a local maxima cannot also be a local minima unless it lies along the surface boundary.

6.4 Envelope Generation

A critical stage in the EMD algorithm is the generation of the extrema envelopes. The envelopes represent the outer and inner bounds of the signal, and need to be as close to the underlying surface as possible to ensure timely convergence. This leads to the question of how to connect the extrema points – which are sparsely sampled, and often clustered in certain regions – in a topologically and geometrically correct fashion. There are many alternatives for connecting a set of sampled data points to form a 3D surface. These techniques can be broken down into 2 categories: interpolation surfacing, and approximation surfacing.

Interpolation surfacing is the process of using known data points in a defined neighborhood to estimate previously unknown surface points. The generated surface passes exactly through the known data points and is interpolated via some function between the data points. There are many algorithms available to interpolate surfaces from a set of data points.

Bernardini et al. (Bernardini et al., 1999) developed a method for interpolating surfaces known as Ball Pivoting. Three points form a triangle if a ball of user-specified radius ρ touches them without containing any other point. In other words, starting with a seed triangle, the ball pivots around a triangle edge until it contacts another point. That point and the endpoints of the pivot edge make up the new triangle. This process continues until all edges have been tried. The radius can be modified to handle disparate sampling densities.

Amenta, Bern, and Kamvysselis (Amenta et al., 1998) developed a surface reconstruction algorithm based on the 3D Voronoi diagram and Delaunay triangulation of a point set. It produces a set of triangles called the *crust* of the sampled points. The crust algorithm provides a surface that is topologically correct, handles varying density of sampling points, and converges to the original surface.

Bajaj, Bernardini, and Xu (Bajaj et al., 1995) developed a surface reconstruction algorithm based on alpha shapes. A signed distance function is defined that gives a positive value if a given query point $\boldsymbol{q}$ lies outside the object, and a negative value if it lies inside the object. Alpha shapes are used to compute a piecewise linear approximation of the zero set of this function. Delaunay triangulations and Clough-Tocher smoothing methods are used to reduce the approximation error and smooth the resulting surfaces.

When the measured data values are accurate, surface interpolation is appropriate. However, experimental data is rarely accurate, and as the uncertainty of the measured points increases, interpolation is no longer ideal because the surface may no longer pass through the measured data points. Also, many interpolation algorithms exhibit unwanted behavior - such as bulls-eye effects and sharp discontinuities - when the data is not uniformly distributed, as is the case with our extrema points. However, there is a solution – approximation surfacing constructs a surface that lies near the sampled data points according to some estimate of the data uncertainty.

Hoppe (Hoppe et al., 1992; Hoppe et al.) used simplicial surface reconstruction, based on point normal evaluation, where an implicit function *f(x)* at each point is evaluated as the distance to the tangent plane at each point. A tessellated surface is then generated using the marching cubes algorithm.(Lorenson & Cline, 1987) He also introduced a subdivision surface modeling scheme to improve performance around boundaries, creases, and other geometrically important features.

A recent advance in surface approximation is Dinh's work on implicit surfacing using volumetric regularization.(Dinh, et al., 2002) Dinh's work uses volumetric regularization to constrain the implicit function representing the surface to lie as close to the sampled data points as possible, while preserving the overall functional smoothness. She extends the current work on volumetric regularization by introducing a basis function with multiple orders of smoothness, providing the ability to generate smooth surfaces for sparse, noisy data sets. Because of this, we have chosen to use Dinh's work as the basis for our envelope generation method.

6.4.1 Volumetric Regularization

With volumetric regularization, the unknown surface functional is found by minimizing the energy function

$$H[f] = \beta[f] + \frac{1}{\lambda}\sum_{i=1}^{n}(y_i - f(x_i))^2 , \qquad (6.2)$$

where β is the desired smoothness functional (such as thin-plates), f is the unknown implicit surface function, n is the number of constraints, the y values are the observed data points, and λ is the *regularization parameter* that acts as a balance between the data fit and the smoothness of the function. For $\lambda > 0$, the surface passes close to, but not necessarily through the observed data points, hence we have surface approximation. As λ approaches 0, the surface passes closer to the measured data points and there is less smoothing of the data.

Derivations given in (Wahba, 1990) show that the cost functional given in (6.2) is minimized by a sum of weighted radial basis functions

$$f(x) = P(x) + \sum_{i=1}^{n} \omega_i \varphi(|x - c_i|). \tag{6.3}$$

In the above equation, *f(x)* is an implicit function, *P(x)* is a polynomial term that spans the null space of *f(x)*, ω are the weights for the basis functions, φ is the radially symmetric basis function, and *c* are the centers of each basis function (constraint points).

The unique implicit function *f(x)* is generated by solving for the unknown weights and the coefficient of the polynomial *P(x)*. These unknowns are determined by constructing and solving the linear system defined by applying (6.3) to every constraint.

$$\begin{bmatrix} \varphi(r_{11})+\lambda_1 & \dots & \varphi(r_{1n}) & 1 & c_1 \\ \vdots & \ddots & \vdots & 1 & \vdots \\ \varphi(r_{n1}) & \dots & \varphi(r_{11})+\lambda_n & 1 & c_n \\ 1 & 1 & 1 & 0 & 0 \\ c_1 & \dots & cn & 0 & 0 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \vdots \\ \omega_n \\ p_0 \\ P \end{bmatrix} = \begin{bmatrix} f(c_1) \\ \vdots \\ f(c_n) \\ 0 \\ 0 \end{bmatrix} \tag{6.4}$$

$$r_{ij} = |c_i - c_j|$$

Given the above formulation of the linear system used to solve for the weights, it can be seen that setting $\lambda_i = 0$ forces the weights to be calculated based exactly on the measured data. Setting all λ_i to 0 has the effect of exact data interpolation, forcing all new points to be constructed as a weighted sum based on the original measured data, yielding a rougher surface. As the λ_i's grow larger, more data smoothing is added to the system. For very large values of λ_i the reconstructed surface is extremely smoothed.

The constraints used to formulate the linear system in (6.4) are a combination of surface and exterior points. The function values at each constraint are known since we have defined the constraint points to lie either on the surface (measured) or outside the surface (assigned). All exterior constraints are placed at the same distance away from the measured surface by simply locating a point in space a specified distance *d* along the vertex normal. These exterior constraints do not represent real data, only an indication of surface orientation.

The external constraints are only needed for an isosurface representation of the 3D surface. For 3D signal (volume) representation, the variation of the measured signal on the 3D domain will provide enough information for the 3D function to be interpolated without need for arbitrarily assigned points to indicate direction of signal flow.

6.4.2 Radial Basis Functions

The radial basis functions (RBFs) used to determine *f(x)* are radially symmetric functions centered at the constraint points. They can be designed to fit any number of specifications, but for our purposes we need the RBFs to exhibit only local influences – i.e., the effect of the

RBF decreases dramatically with distance from the center of the basis function. Dinh has used an RBF with multiple orders of smoothness to ensure a smooth surface fit, while minimizing the effect of the RBF at distance. The smoothness functional used is associated with the following partial differential equation:

$$-\delta\Delta f + \Delta^2 f - \tau\Delta^3 f = 0\,. \tag{6.5}$$

Here, δ controls the amount of first-order smoothness, τ controls the amount of third-order smoothness, and the balance between the two controls the second-order smoothness. The radial basis function that minimizes the above equation is

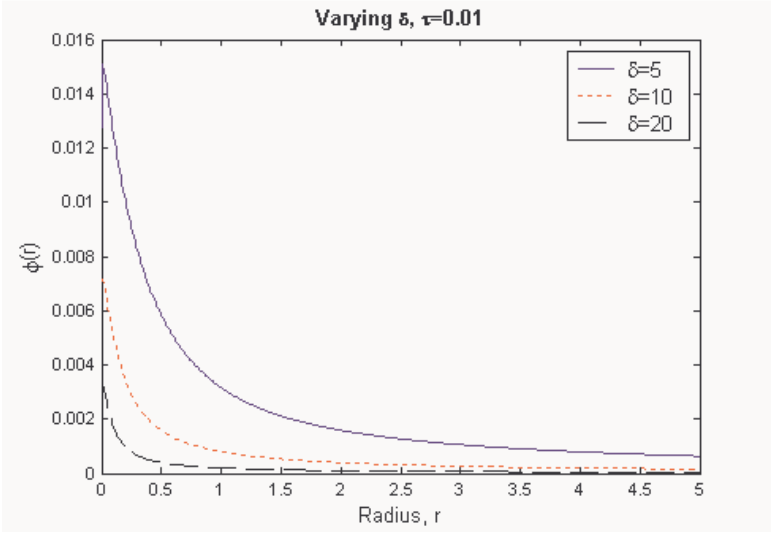
$$\varphi(r) = \frac{1}{4\pi\delta^2 r}\left(1 + \frac{we^{\sqrt{v}r}}{v-w} - \frac{ve^{\sqrt{w}r}}{v-w}\right), \qquad v = \frac{1+\sqrt{1-4\delta^2\tau^2}}{2\tau^2} \quad w = \frac{1-\sqrt{1-4\delta^2\tau^2}}{2\tau^2} \tag{6.6}$$

where r is the Euclidian distance from the function's center to the point of interest. This choice of RBF quickly falls towards zero as r increases, providing the locality of influence that we desire. This local influence makes the linear system of (6.4) a sparse matrix, which can be quickly solved for using fast, sparse matrix solvers such as the *conjugate gradients squared* method.

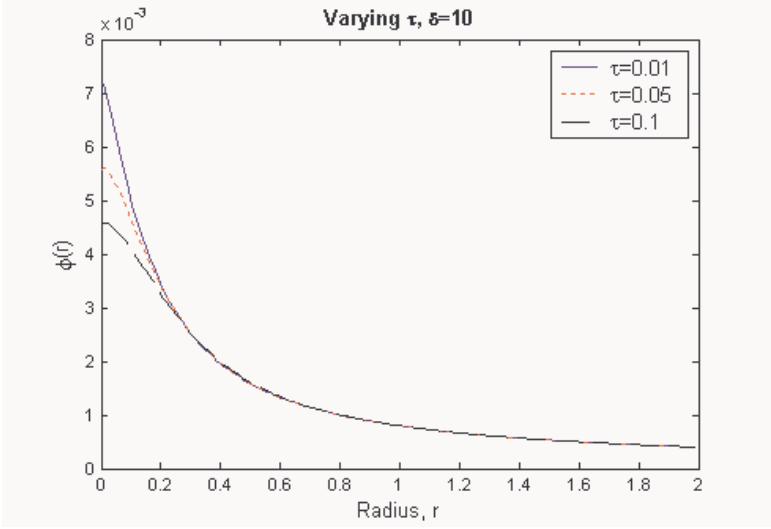
Figure 6.8 shows how the choice of the smoothness parameters δ and τ affect the overall influence of the radial basis function. As δ increases, the region on which the RBF exerts influence shrinks. As τ increases, the curvature continuity of the RBF is preserved, at the cost of broadening the influence of the RBF around its center. In both plots. It is evident that the RBF value approaches zero as the distance from its center grows large. The static values in each of the plots (δ = 10 and τ = 0.01) are the values that will be used in the rest of this paper to generate the results shown.

6.4.3 Envelope Generation

To generate the extrema envelopes, we use the maxima and minima points extracted by the means discussed in Section 6.2 as surface constraints, and define exterior constraints about these points to define surface orientation. For each envelope, there are $1/10^{th}$ as many exterior constraints chosen as surface constraints. The exterior constraints are chosen as points an arbitrary distance from a random sampling of the surface constraints. The linear system given in (6.4) is constructed form these constraints and the implicit surface *f(x)* is generated.



(a)



(b)

Figure 6.8. Demonstration of how the choice of smoothness parameters affects the radial basis function. (a) Cross section of the RBF with varying values of δ. (b) Cross section of the RBF with varying values of τ.

Figure 6.9 shows the envelope generation process for the minima envelope of the curvature computed on our example object. The constraints used to develop the surface are shown in Figure 6.9a, with the 1324 surface constraints shown in purple, and the 132 exterior constraints shown in gray. A marching cubes-rendered version of the minima envelope is shown in Figure 6.9b. The voxel size of the cubes was set to be 0.006% of the length of the object.

6.5 3D EMD Algorithm

The algorithm for 3D Empirical Mode Decomposition is a direct extension of the original EMD for 1D signals. We define the 3D sifting process as:

1. Identify the signal extrema of the signal X through the definition of a geodesic neighborhood on a tessellated mesh, or through a $\boldsymbol{\delta}$-ball approach to neighborhood assignments for non-tessellated 3D domains (point clouds);
2. Generate the 3D signal extrema envelopes by connecting maxima (and minima, respectively) points with a RBF;
3. Calculate the local signal mean m_j by averaging the values of the maxima and minima envelopes;
4. Subtract the mean from the signal: $X - m_j = h_j$ to yield a zero local mean; and
5. Repeat on h_j until h_j is an IMF.

(a) (b)

Figure 6.9. Minima envelope generation for the waterneck model. (a) Surface(purple) and exterior(grey) constraints used to generate the surface. (b) The implicit surface of the minima envelope, rendered using marching cubes with a voxel size of 0.03 mm^3.

As in the 1D case, the process is sifting out the first local frequency mode from the signal on the 3D domain (e.g., mesh) based only on the highest characteristic multiscale frequency present in the signal. The sifting process continues until riding waves have been eliminated and uneven amplitudes have been removed. This yields the first IMF, the highest spatial frequency component in the signal. Once the first IMF has been identified, it is removed from the signal X and the sifting process is begun anew on the residual. This process is continued until there are all characteristic scales of the data (IMFs) have been isolated.

The algorithm for sifting the IMFs from signal $X \in M(\Re^3)$ is as follows:

1. Initialize $R_0 = X$ (the residual) and $j = 1$ (IMF index number),
2. Extract the jth IMF;
    a. Initialize $h_o = R_{j-1}$, i=1,
    b. Extract local extrema of h_{i-1},
    c. Compute maxima and minima envelopes E_{max} and E_{min} by interpolating, respective the local maxima and minima of h_{i-1},
    d. Compute $m_{i-1} = (E_{max} + E_{min})/2$ (mean envelope),
    e. Update $h_i = h_{i-1} - m_{i-1}$ and $I = i+1$, and
    f. Calculate stopping criterion based on standard deviation (SD) of the difference between h_i and h_{i-1}.
    g. Repeat steps (b) to (f) until stopping criterion is met - also known as sifting - and set $IMF_j = h_i$
3. Update residual $R_j = R_{j-1} - IMF_j$; and
4. Repeat steps (2) to (3) until the number of extrema in R_j is less than 2.

6.6 3DEMD Results

We have performed a number of experiments to demonstrate the performance of our extension of EMD to 3D surfaces. These experiments involve synthetic and real (digitally acquired) 3D surfaces with associated synthetic and real (measured or computed) surface attributes. The test models range from simple objects to complex, multi-apart objects, while the attribute signals range from simple compositions to complex functional evaluations along the surface. The purpose of this wide variety of test objects and evaluation signals is to demonstrate the general performance of our algorithm, and to show that it has applicability in multiple fields of 3D signal analysis. For the remainder of this section, we refer to a synthetic signal as one that has been arbitrarily assigned to the surface, and a real signal as one that has either been measured (e.g., temperature, pressure, etc.) or derived from the surface geometry (e.g., curvature, centricity, etc.).

The experiments begin with a simple synthetic function evaluated on a simple surface. Figure 6.10 shows a model of a sphere consisting of nearly 10K points and 18K triangles.

The function evaluated on the model is a sum of sinusoids, based on the *z*-value of the vertices. Specifically, the surface attribute is defined as $f(x, y, z) = 1.4\sin(3\pi z / 2) - 0.75\sin(\pi z / 2.25) + 1.3$. The 3DEMD algorithm was used to decompose this sinusoidal signal on the sphere into two IMFs plus a residual component. The first IMF captures the highest frequency band of signal oscillation, while the residual captures the overall trend of attribute along the 3D surface. These results correspond to those obtained from the 1D and 2D implementations of EMD.

The next experiment involves analyzing the maximum surface curvature of the sphere model. Each vertex has been assigned the maximum curvature of the surface evaluated at that vertex using Taubin's method (Taubin, 1995a). Ideally the surface curvature of a sphere should be constant, but the discretization process involved in generating a triangular mesh of a sphere yields small variations about this constant. Figure 6.11 shows the 3DEMD results of the sphere curvature model. The decomposition yields expected results. The first IMF isolates the highest frequency bands of signal oscillation, leaving the Stage 1 residue to contain a "smoother" version of the curvature, with the highest frequency variations removed. The full decomposition consists of 6 IMFs and a residue. The final residue contains the overall trend, or bias, of the curvature measurements. The average value of this trend is 1.69, and compares favorably with the theoretical curvature value of 1.67 for a sphere of this size (radius =0.6).

Next is an analysis of curvature on a more complex model. Here, a waterneck has been digitized and its surface curvature computed and attached to each vertex as a feature attribute. This model contains approximately 15K vertices. The full analysis yielded 11 IMFs plus a residual. The top section of Figure 6.12 shows the original model and several IMFs during the 3DEMD process. The bottom row of Figure 6.12 shows the 3DEMD process centered around a single vertex, located on the mounting bolt. The intent here is to show that both globally and locally, the 3DEMD process is isolating the highest oscillating frequencies in every IMF, leaving behind the lower frequency information for further analysis stages.

To demonstrate the use of 3DEMD on a point cloud, all edge information was removed from the waterneck model. The original surface curvature signal was used as the feature for analysis, having the tessellated decomposition for comparison. For the neighborhood range, $\boldsymbol{\delta}$ was chosen to be equal to the average edge length of the original tessellated mesh. Figure 6.13 shows the results of the 3DEMD decomposition of this point set. The top section shows the original signal assigned to the 3D points, along with several of the IMFs. As before, the bottom row shows the results of all IMFs centered around a single point located on the mounting bolt.

Comparing these results with that from the tessellated case, it can be seen that the IMFs generated by both methods are very similar in magnitude and form. The differences between them can be attributed to the different ways the vertex neighborhoods were calculated and thus the definition and identification of the local extrema. One observation to note is that the tessellated model was nearly regularly sampled, meaning that choosing the average edge

length for the local neighborhood δ-ball was a logical choice. If the original tessellation had been very anisotropically sampled, the result would likely not have been as similar.

Perhaps one of the most illuminating aspects of the 3D EMD is the ability to decompose and analyze measured signals on a 3D surface model. Figure 6.14 shows this process on a fan blade model with pressure measurements attached to the surface. This model was taken from the online demonstration datasets from the INUS Technologies Rapidform 3D software website (INUS Technologies, 2007). The full decomposition yielded 9 IMFs plus a residual. Here the 3D EMD analysis performed well, even though the location of the measurements on the surface were not evenly distributed. In fact, the sampling density on the "complex" part of the fan blade (the mounting area) is roughly 4 times higher than that of the blade part. The anisotropic sampling and chaotic nature of the pressure measurements contributed to this analysis having the longest running times yet, with the IMFs requiring an average of 227 sifting operations per IMF compared with the previous average of less than 100 for each of the previous 3D models.

For large scale mobile scanning data, there are a number of signals that can derived from the scanning process and 3D geometry, including: estimated noise, range values, curvature, and frequency of deviation, among others. For experimental purposes, we have chosen to perform the 3D EMD on the curvature estimated from the 3D positions of the processed model. The appropriateness of estimating curvature on a discrete, noisy surface such as that generated from a mobile scanning system has been called into question in the past, as an inelegant approximation that does not strictly adhere to the analytical bounds of such second order functions. However, experimentally, curvature estimation has proven to be a reliable tool for a variety off application, including smoothing and denoising(Desbrun et al., 1999; Sun, et al., 2002; Ki et al., 2004).

Figure 6.15 shows the Strip Mall data acquired by our system and shown in Figure 3.5, with the surface curvatures associated to each vertex. As before, the signal attached to the 3D model is the maximum vertex curvature, as computed by Taubin's method. The high variation of curvatures on the surface implies a noisy model, since we know that the building surfaces themselves should be regular structures with smooth surfaces - as observed from physical inspection. From an analysis standpoint, the curvature of the model is interesting, as it describes the variations of the measured surface. In the context of spatio-temporal analysis, the variations of the curvature signal across the mesh can give a better idea of the underlying surface and modeling noise issues.

Figure 6.16 shows the results of performing our 3D EMD algorithm to the 3D model above. The full decomposition yielded 10 IMFs. A zoomed in region of the model can be seen in Figure 6.17 showing a close up view of a small region as the IMF extraction process progresses. It is promising to note that even on this larger-scale version of data (approximately 350,000 vertices) the algorithm iterated to a solution. The sampling density for this model is roughly 3 times greater in the vertical direction than the horizontal, once again demonstrating the effectiveness of the 3D EMD to real world data. Regarding execution, the IMFs for this model required an average of 90 sifting operations per IMF.

At this point, it is important to note that the 3DEMD results at each level of the signal decomposition exist at every vertex in the original model. The 3D geometry is not changed in the decomposition process. This is in contrast to the existing 3D decomposition methods discussed in Section 2.4, where the arbitrary removal of some vertices at every decomposition level yielded no information from those vertices at lower decomposition levels. This information is important for many signal processing applications (Huang, et al., 1998). As such, it is impossible to directly compare the information and analysis yielded by the 3DEMD approach to those decomposition methods presented in Section 2.4.

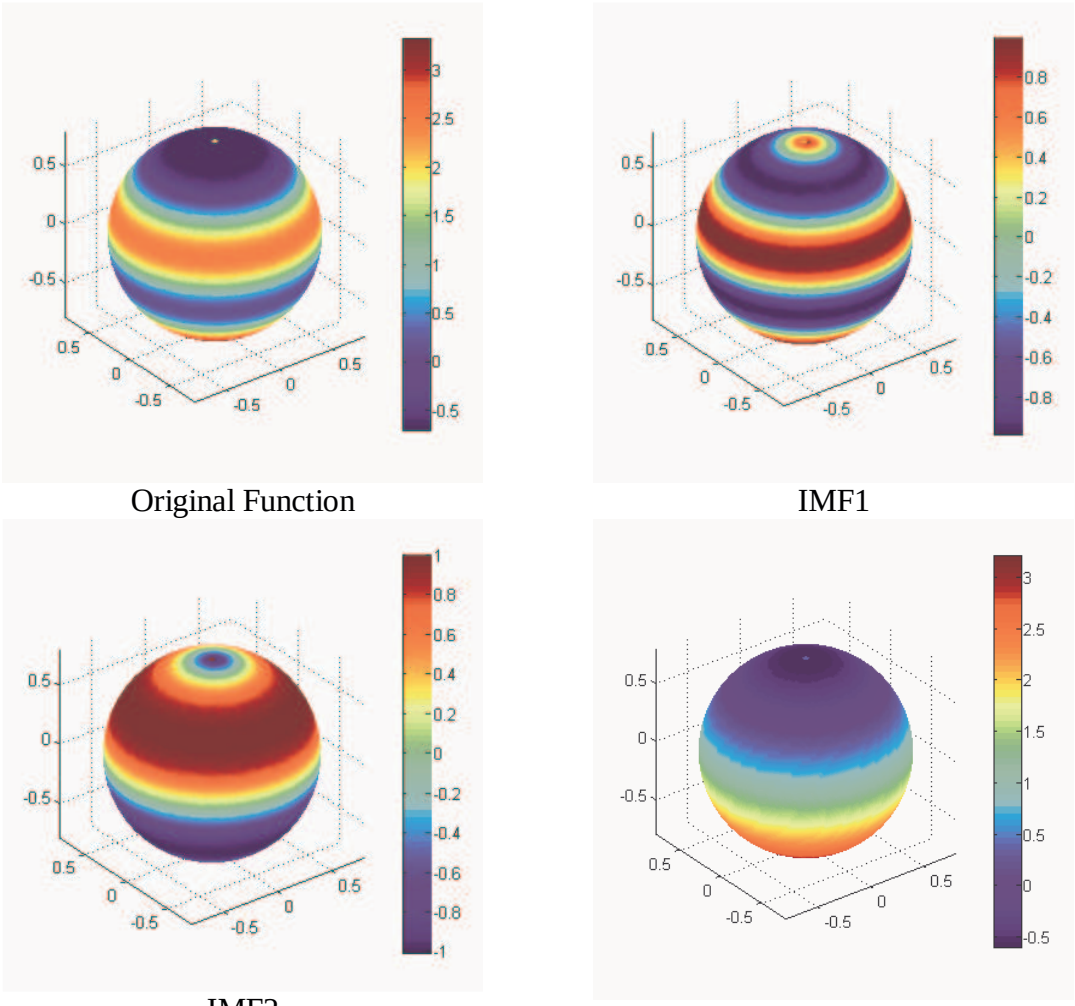


Figure 6.10. Decomposition of a sinusoidal signal evaluated on a sphere.

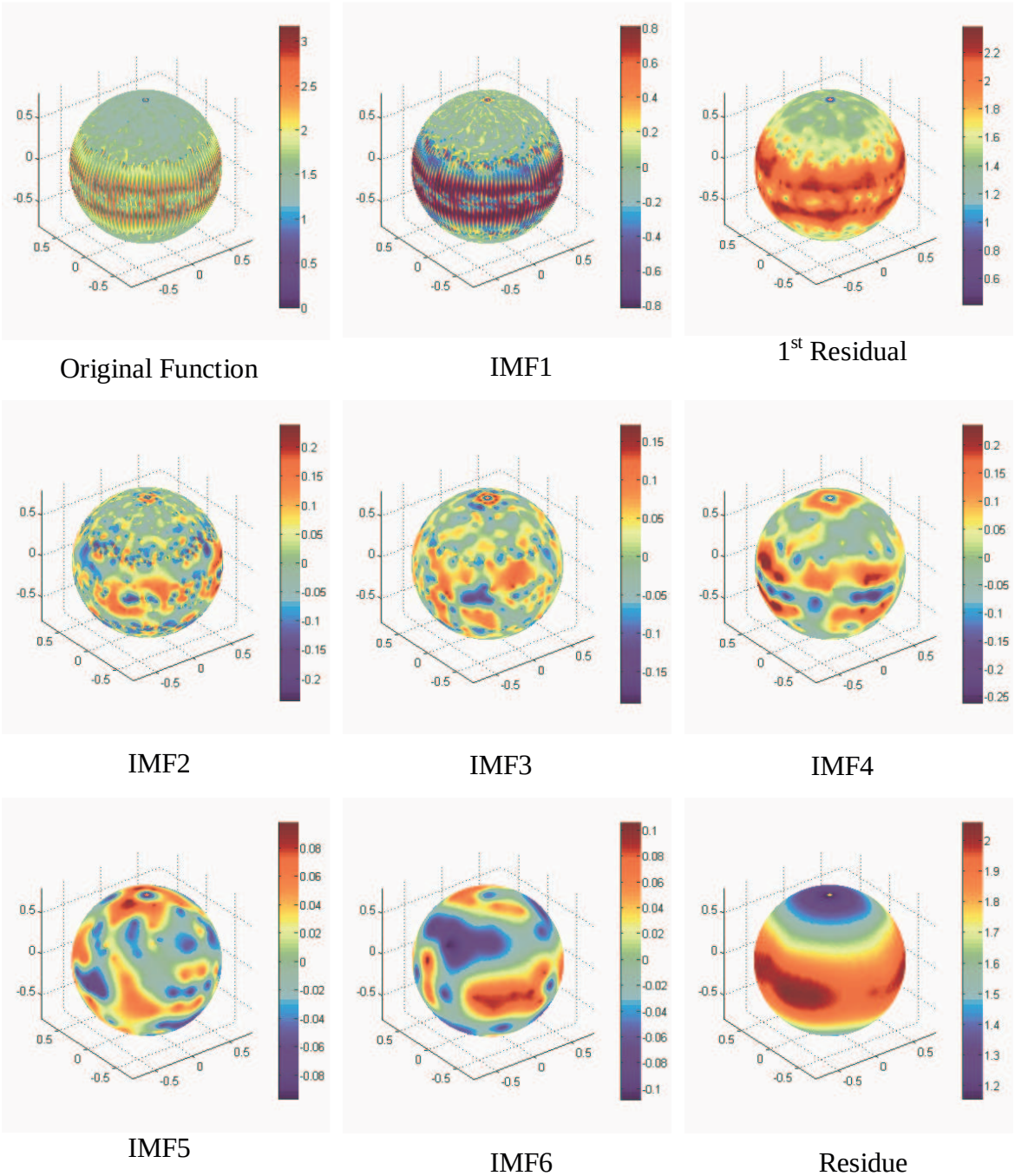


Figure 6.11. 3DEMD results on the maximum surface curvature evaluated on a sphere.

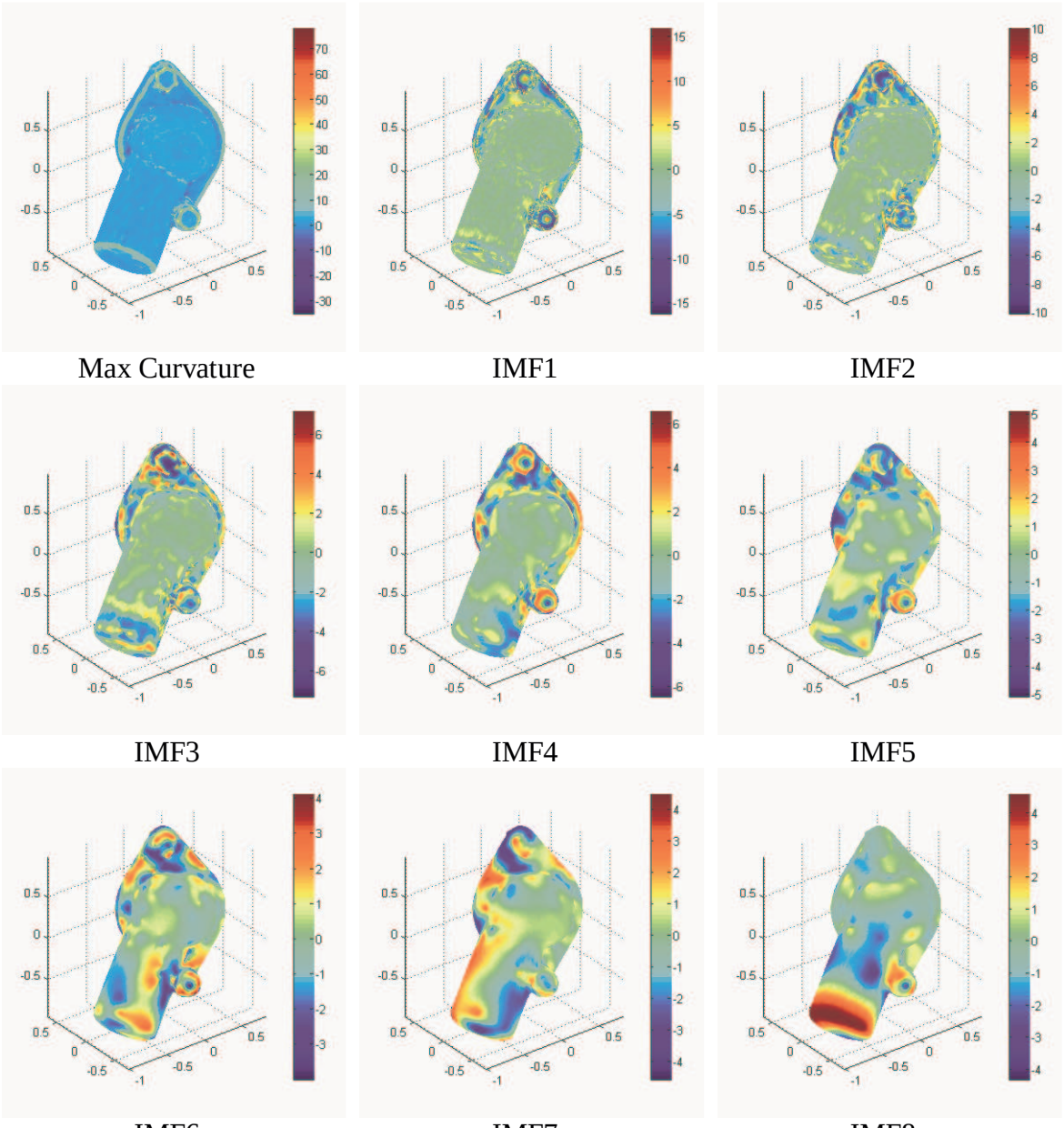
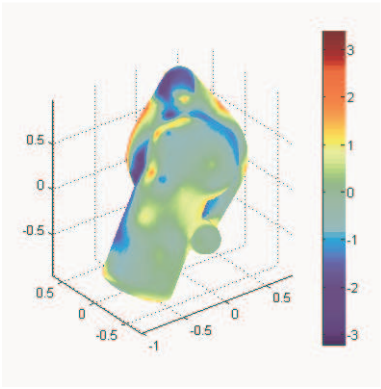
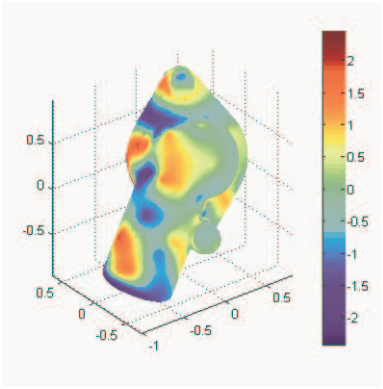


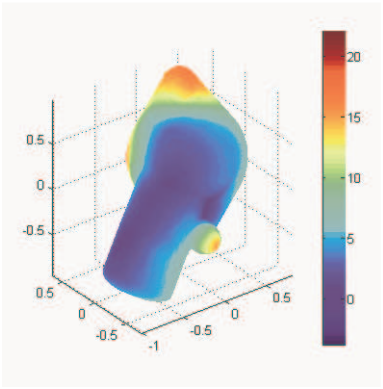
Figure 6.12. 3DEMD results on a model of a waterneck with the maximum curvature analyzed on the surface. The top shows a series of IMFs over the entire model. The bottom row shows detailed views of one portion of the model as it progresses through all 11 IMFs.



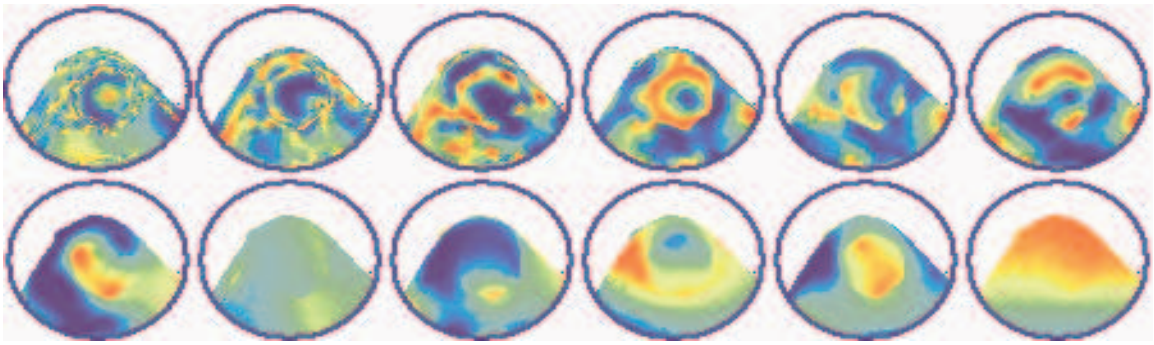
IMF9



IMF10



Residual



IMF Progression

Figure 6.12. Continued.

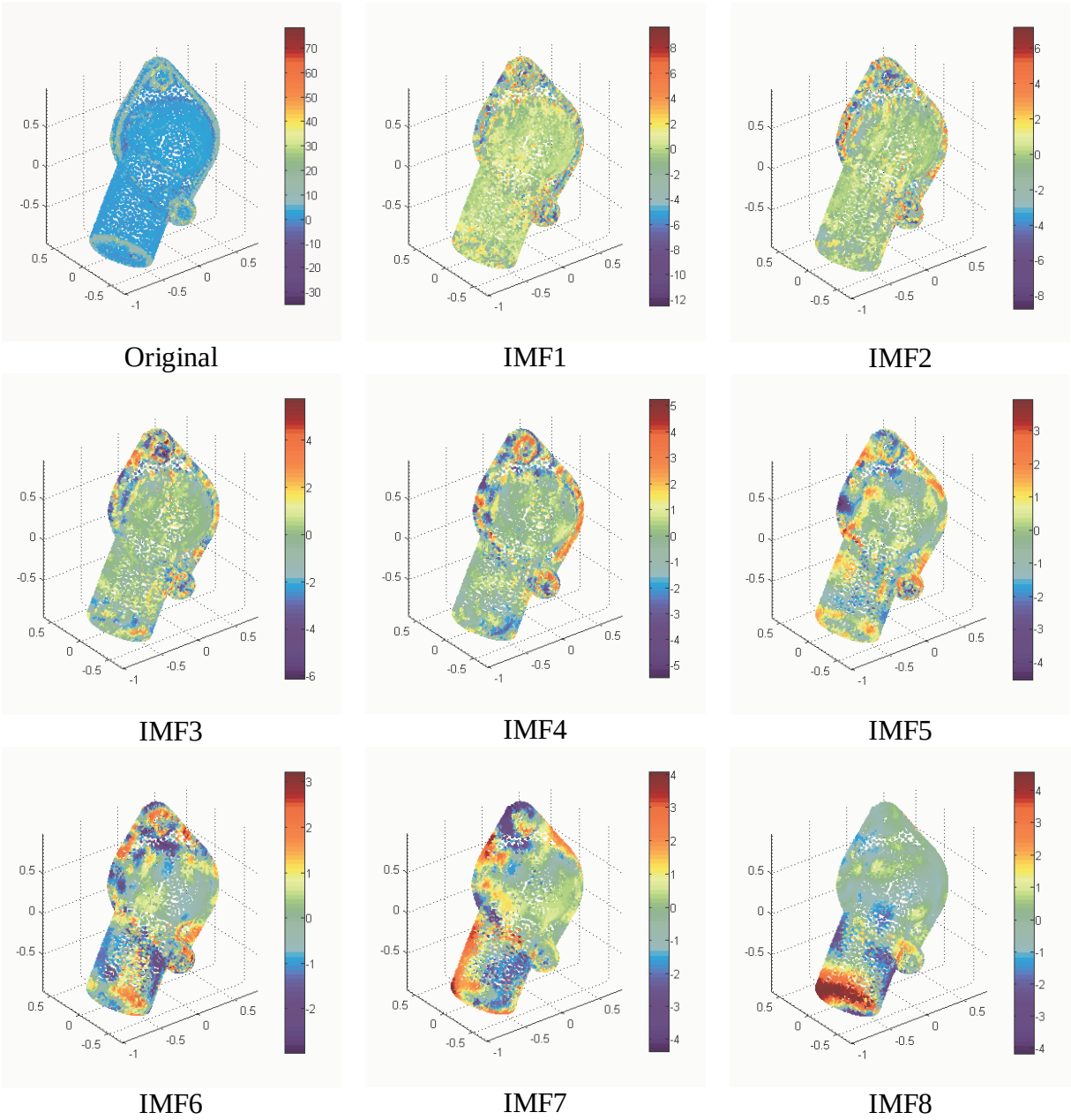
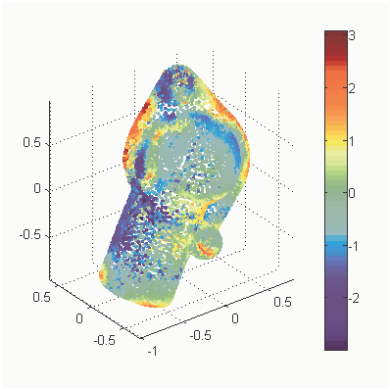
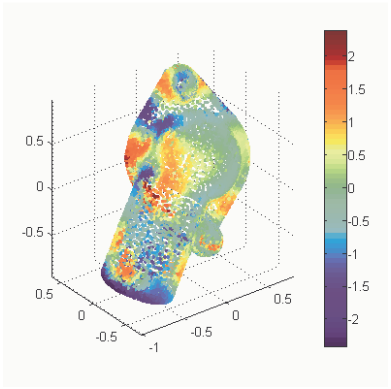


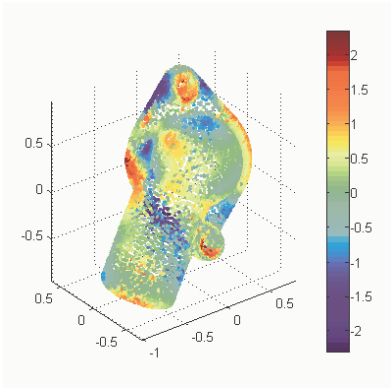
Figure 6.13. 3DEMD results on a point cloud representation of the waterneck model, with the surface curvature associated with vertices as before. The top shows a series of IMFs over the entire model. The bottom row shows detailed views of one portion of the model as it progresses through all the IMFs.



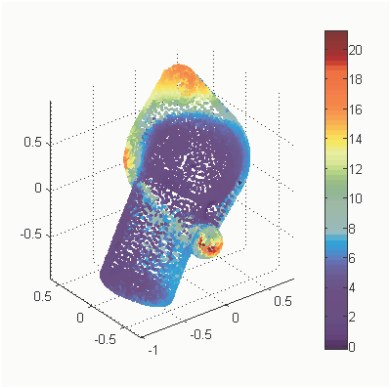
IMF9



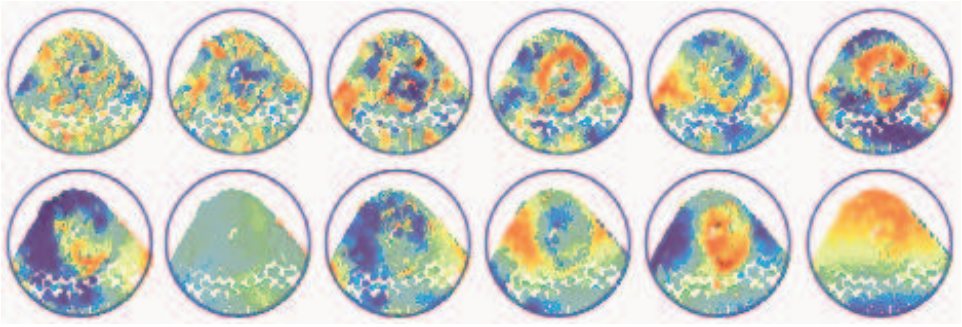
IMF10



IMF11



Residue



IMF Progression

Figure 6.13. Continued

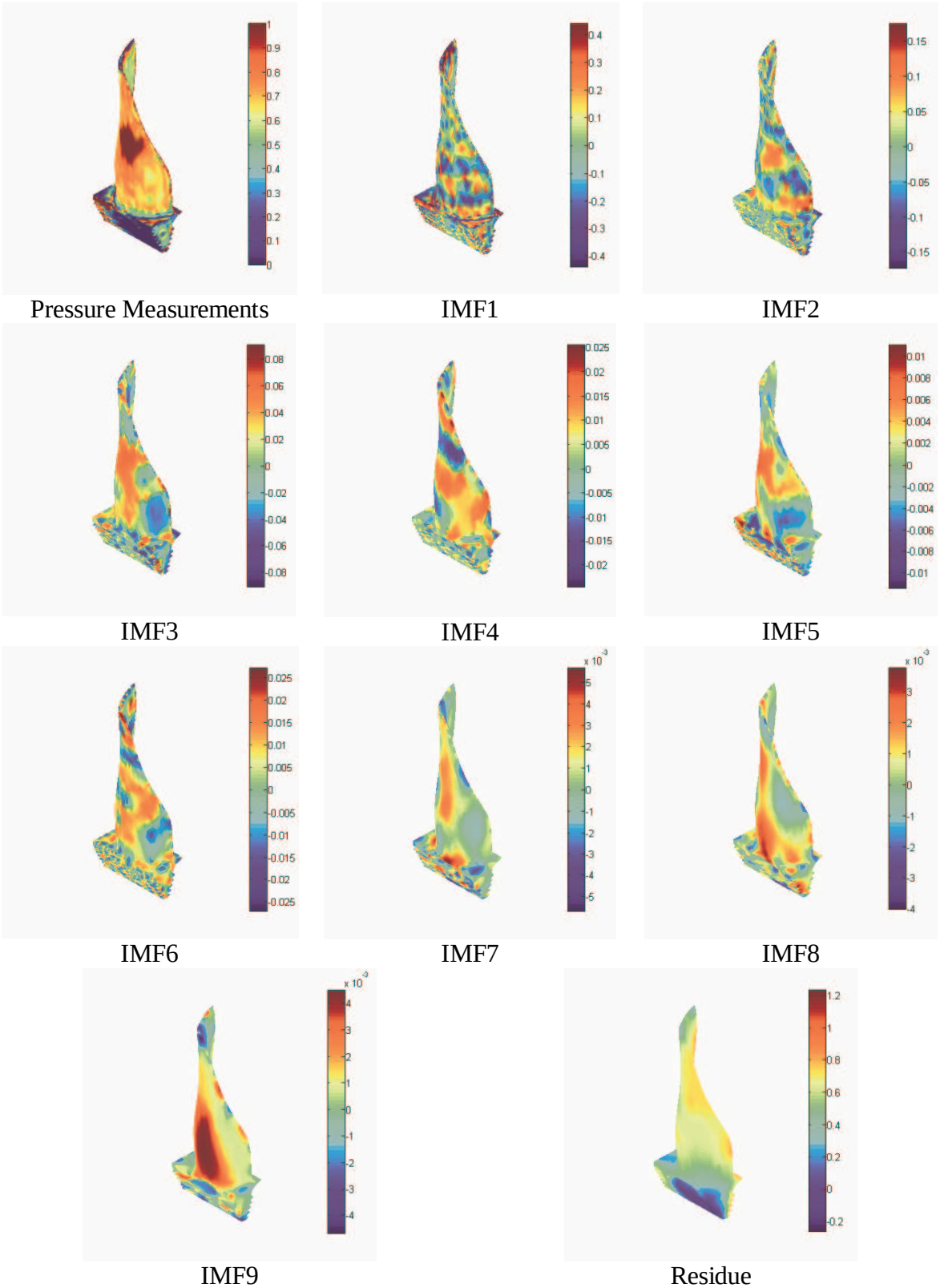


Figure 6.14. 3D EMD decomposition of a pressure analysis of a fan blade model.

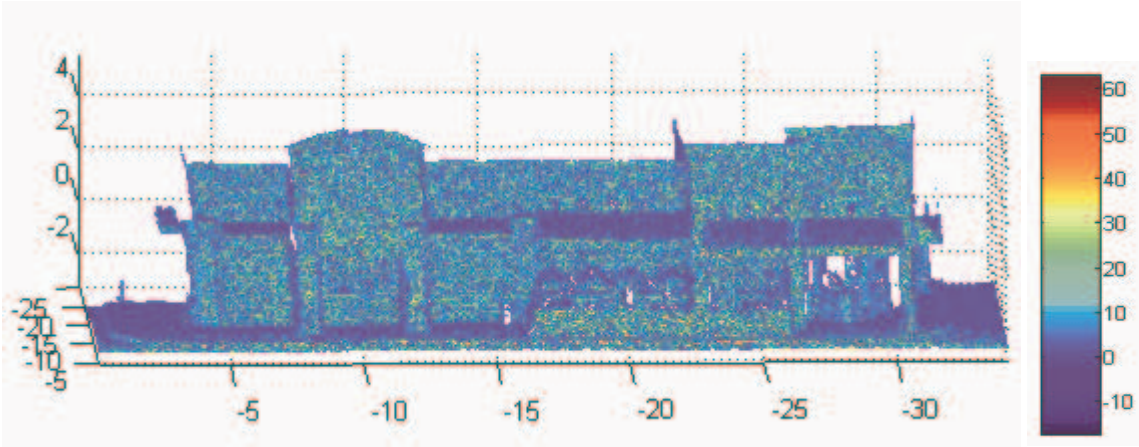


Figure 6.15. Strip Mall dataset with every vertex colored according to its maximum curvature.

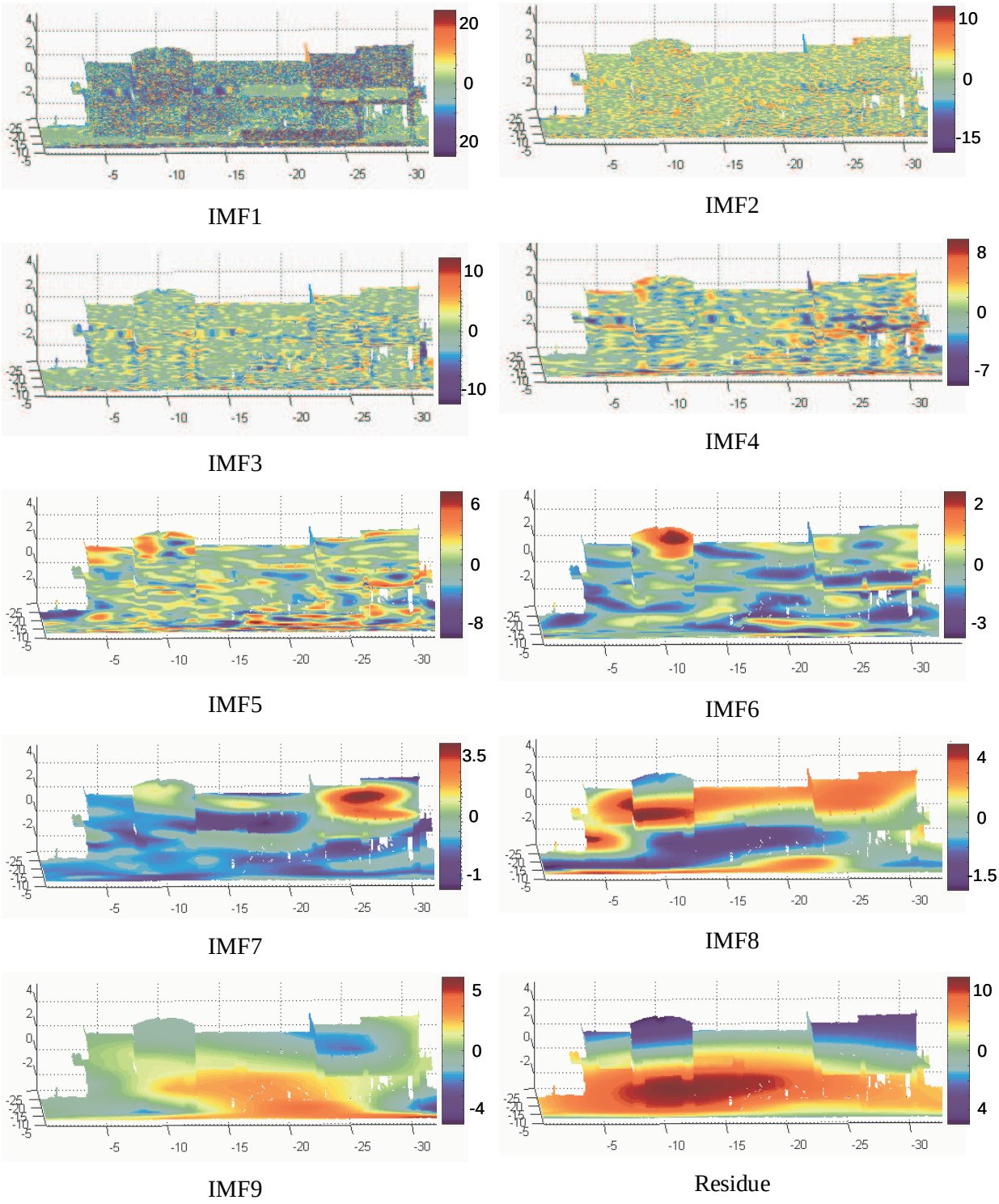


Figure 6.16. IMFs extracted during 3DEMD analysis of the Strip Mall dataset with maximum curvature as the surface value.

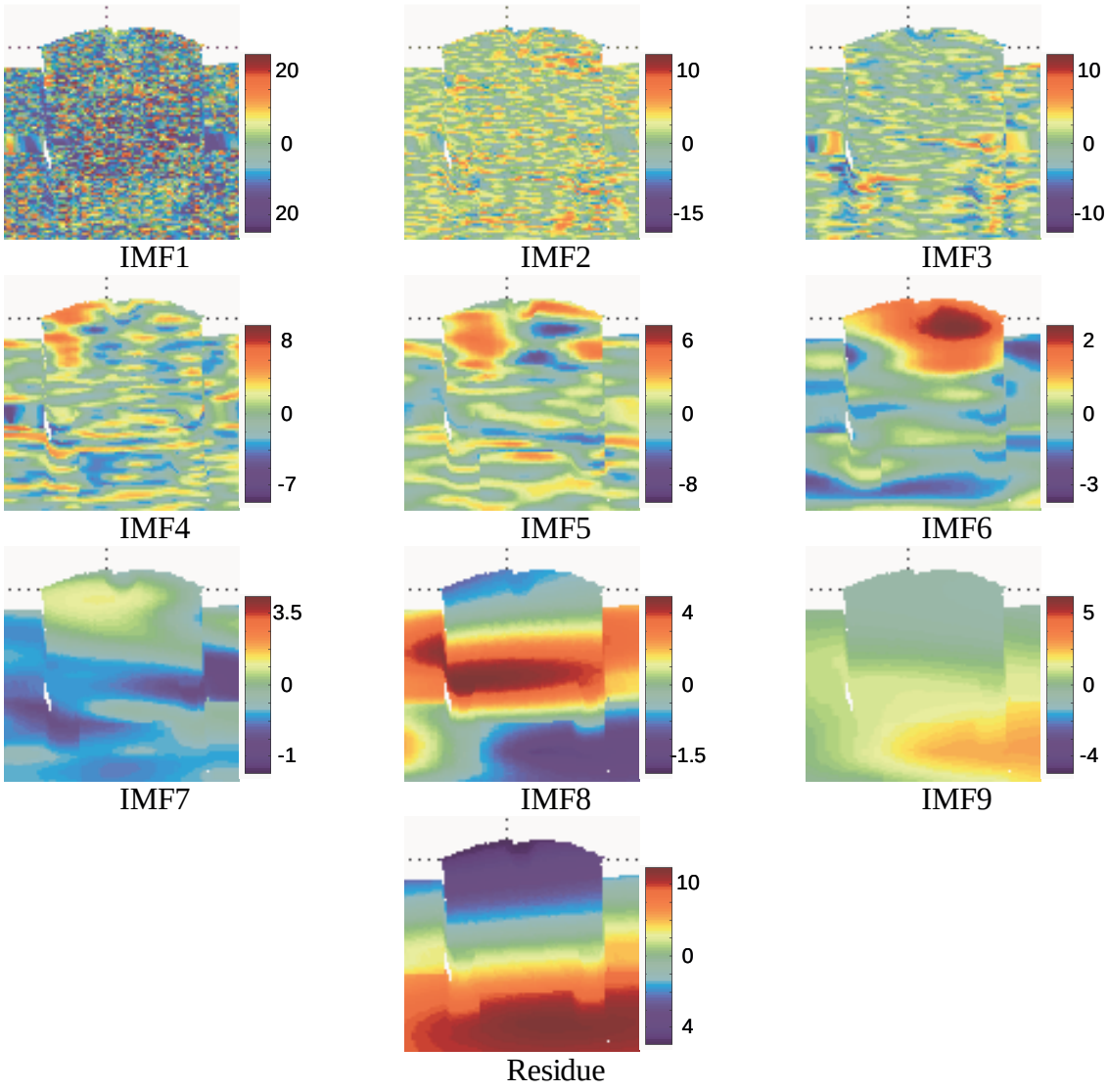


Figure 6.17. Zoomed in region of the dataset shown in Figure 6.16 as it progresses through all 10 IMFs.

Table 6.1. Summary of 3DEMD results by dataset.

Dataset - Attribute	# of points (approx.)	# of IMFs	Avg. Iterations/IMF
Sphere - composition	12,000	3	9
Sphere - curvature	12,000	7	10
Waterneck - curvature	15,000	11	58
Waterneck - point cloud	15,000	11	62
Fanblade - pressure	21,000	10	227
Strip Mall - curvature	150,000	10	91

7 Application: Source Separation

We believe that the basis for 3D surface signal analysis provided by the 3DEMD decomposition will prove beneficial for a variety of 3D modeling and analysis applications. In order to demonstrate the flexibility of the analysis, we have chosen to use 3DEMD as the basis for a novel algorithm for separating the feature space of a 3D model into individual source signals, either for identification of source or for further processing.

The concept of source separation revolves around the premise that the data being observed has information from a number of signals mixed together, with the goal of identifying the data influence of the individual sources. Source separation can take into account *a priori* knowledge of the representative signals, yielding a tuned source separation algorithm. Alternatively, no *a priori* knowledge can be used for the process, which is known as blind source separation.

The very nature of Empirical Mode Decomposition - and 3DEMD by extension - readily lends itself to the separation of source signals, by extracting frequency bands present in the data *without* restriction to arbitrary frequency ranges or basis functions. The capability provided by EMD allows us to identify and separate signals that have disparate underlying frequency responses, without needing to know the frequency response in advance.

Take, for example, the analysis of Figure 6.10. Here, a high frequency signal was identified and extracted as an IMF (IMF1), separating it from the lower frequency components. Indeed, for a dataset where the mixed signals are present in different frequency bands EMD can be viewed as multiple observations of the same signal at different time scales.

To demonstrate the capabilities of 3DEMD in the area of blind source separation under a more realistic scenario, let the 3D mesh to be analyzed consist of the waterneck model with an arbitrary signal on the surface as shown on the top row of Figure 7.1. To this original model, Gaussian noise was added, with amplitude equal to 1/10 the overall range of the underlying signal, zero mean, and unit variance. The corrupted signal is shown on the bottom row of Figure 7.1.

3DEMD analysis of the noisy model was performed and the first IMF is shown on the top row of Figure 7.2. From observation, it appears that the high-frequency additive noise was identified and separated during the decomposition. The upper right image in Figure 7.2 shows the 3D mesh after the first IMF (noise) component has been removed - the first

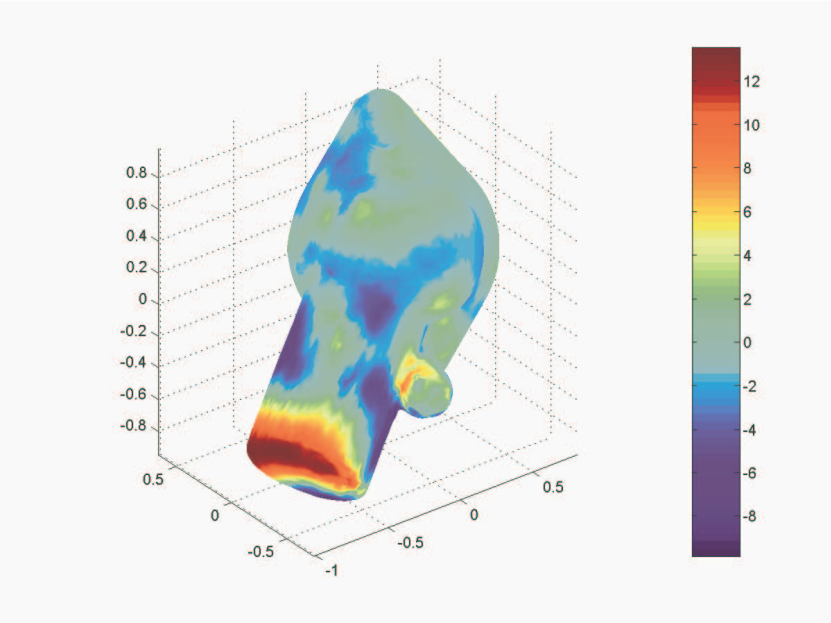
approximation. Comparing the results of the first level of decomposition with our original (noise-free) model, we can see that the first IMF captured virtually all of the noise component. The middle image of Figure 7.2 shows a Laplacian smoothed version of the noisy mesh., while the bottom row shows the deviations of the first decomposition level and the Laplacian smoothed version with the original model, as mapped to the 3D mesh.

Aside from the visual comparison, the quantitative results of the 3DEMD source separation demonstrated in this example are also telling. Table 7.1 shows the minimum, maximum, mean, and median of the difference between the first residue and the original signal. Also shown for comparison are the same numbers for a Laplacian smoothed version of the noisy mesh signal. From these numbers, it is clear that the first IMF captured a significant amount of the distortion, and separating that distortion from the signal yielded a much better representation of the underlying signal than smoothing alone provided.

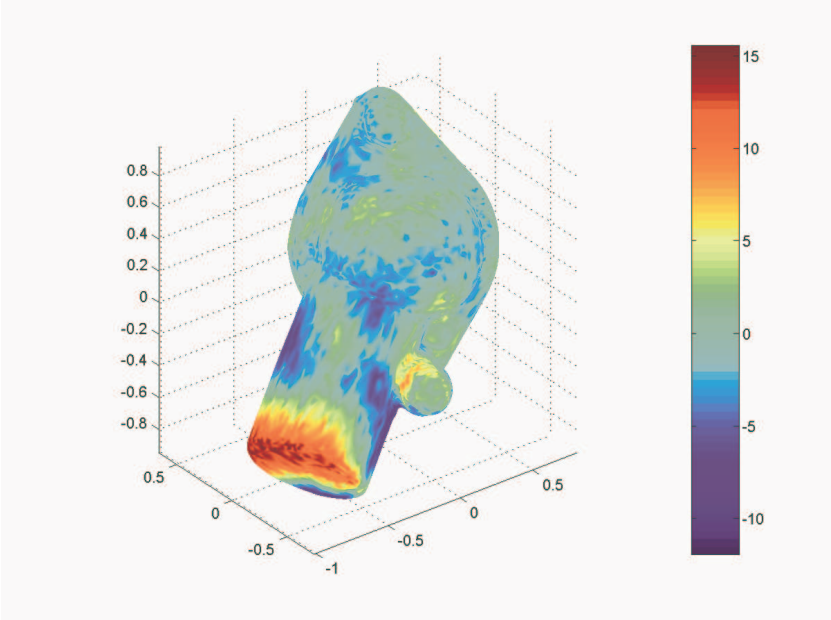
This example illustrates the efficacy of the 3DEMD method for blind source separation of high-frequency noise from a signal measured on a 3D mesh. This straightforward performance of source separation holds only when the desired signal occurs in different frequency bands than the distortion component. When the mixture of distortion and signal covers the same frequency bands, and are thus blended in the IMFs, a more typical blind source separation procedure could be used on the IMFs, as described in (Kamath et al., 2006) for 1D signals.

Table 7.1. Quantitative evaluation of blind source separation using 3DEMD, as compared to Laplacian smoothing. Statistics shown are the Minimum, Maximum, Mean, and Median errors along with the Standard Deviation (SD).

Model	Method	Min	Max	Mean	Median	SD
Waterneck	3DEMD	6.92e-2	-6.93e-2	-3.53e-4	-4.26e-4	3.72e-1
	Laplacian	3.01	-3.31	4.62e-3	2.84e-3	5.31e-1
	Noisy Version	3.96	-3.79	2.73e-3	1.23e-2	1.01

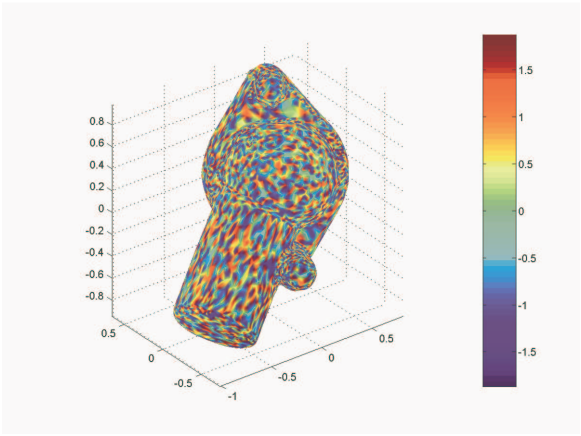


Original (synthetic) function

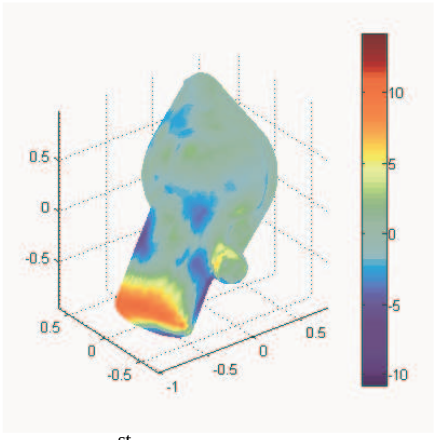


Noisy version

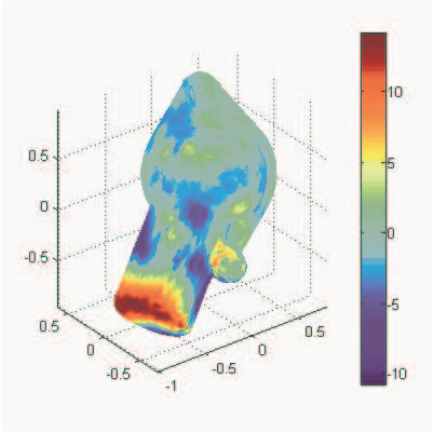
Figure 7.1. Waterneck mesh with an arbitrary smooth function applied (top), and a noisy version of the same signal (bottom).



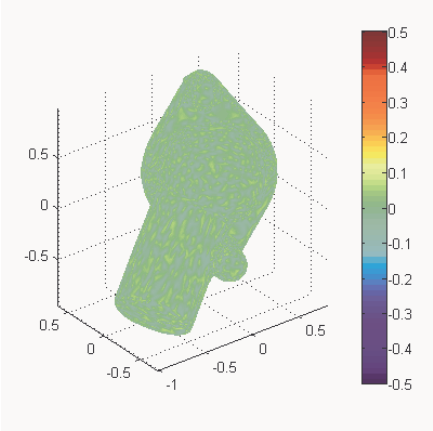
IMF1



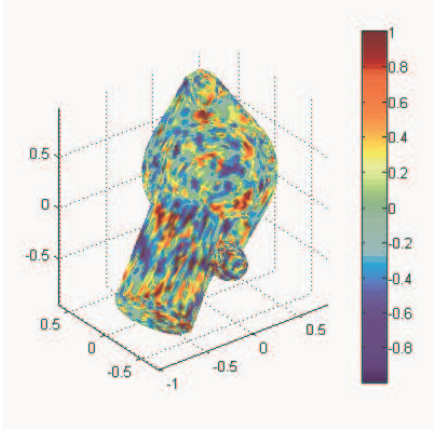
1st Approximation



Laplacian Smoothed



EMD Difference



Smoothing Difference

Figure 7.2. First IMF of the 3DEMD of the noisy waterneck model shown in Figure 7.1, the residue after the first IMF has been extracted from the signal, a Laplacian smoothed version of the noisy model, and the deviations of the residue and smoothed versions from the original signal.

8 Application: Detail Enhancing Denoising

As a second demonstration of the potential of 3DEMD as a tool for the analysis and understanding of 3D meshes and signals, we investigated its potential towards detail enhancing denoising of a 3D surface. In detail enhancing denoising, the surface is analyzed, the noise component(s) identified, and the noise removed to yield a high fidelity surface. Then, the resulting 3D surface is enhanced in the areas of interest to provide a feature rich, high density product. In this application, the surface feature analyzed by the 3DEMD is derived from geometric properties of the 3D mesh. The resulting IMFs are then used to denoise the surface.

This method contrasts with previous denoising and surface smoothing methods in that the surface noise reduction is not based on an *a priori* model of the noise itself, nor is the smoothing done at all scales. Rather, the key point of this denoising method is that we identify the high frequency band noise through source separation by 3DEMD analysis and perform the noise reduction processing only on that component. This denoised version will have the advantages of a smoothed model, with the noise impact removed or greatly reduced, while the regions of high detail (logos, hard edges, etc.) will be enhanced and protected from the smoothing process.

Rather than developing a denoising algorithm based specifically around the 3DEMD analysis, we decided to leverage the wealth of existing 3D surface smoothing algorithms available. The criteria for finding an appropriate existing 3D smoothing algorithm is one that uses a surface-related descriptor as the smoothing driver, can easily be modified to accept the 3DEMD decomposition as the smoothing functional, and one that either is adaptive by design, or can be adaptive based on the IMF decomposition of the curvature. Amongst the algorithms that fit this criteria, we decided to go with the curvature-based regularization method of Ki, et al. (2004). This algorithm uses a combination of area decreasing flow and a median constraint to reduce the effects of both salt and pepper noise, as well as the typical Gaussian noise associated with most 3D digitization systems.

The mathematical model for surface digitization is written as

$$g = f + n, \tag{8.1}$$

where g is the observed 3D surface, f is the true underlying surface, and n is the noise corrupting the digitized model. The optimization problem is therefore one of estimating the underlying surface under the condition

$$O(f) = \|g - f\|^2 - \lambda \|Cf\|^2, \tag{8.2}$$

where C is the filter system applied to f and λ represents a regularization parameter that controls the fidelity to the original surface and smoothness of the restored surface.

The regularized energy functional proposed by Ki et al. combines both an area decreasing flow term and a median constraint for removal of impulse noise. The full regularization form is formulated as

$$f(v) = \sum_{N} \|v - p\|^2 \cos^2 \varphi + \lambda_A h_{area} + \lambda_M h_{median}, \tag{8.3}$$

where N is the neighborhood around vertex v, p is the observed position of v, φ is the approximate angle between the surface and the observation direction - this represents a confidence metric on the measurement and is approximated as the curvature angle at v - and λ_A and λ_M are the regularization parameters that correspond to the area minimization and median constraints respectively. The parameters h_{area} and h_{median} are the surface area for the faces contained in the neighborhood of v and the median deviation of v and its neighbors from the best-fit plane ascribed to them. For a complete discussion on this regularization method, please see *(Ki, et al., 2004)*.

Under the formulation given by Ki et al., the area regularization term in (8.3) was either chosen empirically, or adaptively weighted through the surface curvature. The adaptive weighting had the advantage of preserving the strong edges present in the data, while smoothing out the noise terms. For the 3DEMD, we adaptively choose the regularization term based on the IMF of the frequency band of signals we are attempting to remove. For the denoising case, we typically use the first IMF as the regularization descriptor. Thus, the regularization term for the area decreasing smoothness is formulated as

$$\lambda_A = \frac{\| IMF_v \|}{1 + \theta K_v^{\,2}}, \tag{8.4}$$

where IMF_v is the IMF value at vertex v, K_v is the Gaussian curvature at v, and $0 < \theta < 1$ is a term that represents the sensitivity to 3D edge strength.

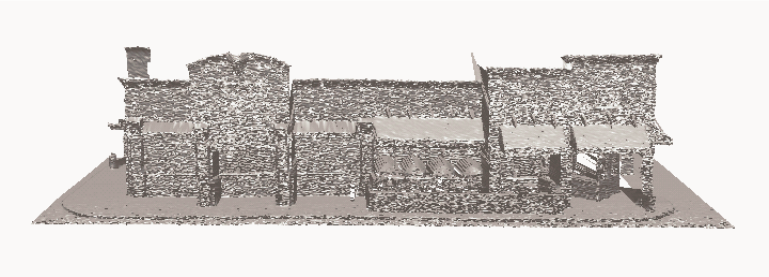
To demonstrate the efficacy of 3DEMD-based denoising on real data, we applied the method to the Strip Mall dataset as decomposed in Figures 6.5.6 and 6.5.7. The top image in Figure 8.1 shows the original, untextured geometry data for the portion of the Strip Mall dataset used in this thesis for demonstrations. Below that is the results of performing the 3DEMD-based denoising. The 3DEMD-based denoising algorithm maintains the fidelity of the edge information, while smoothing out the high-spatial frequency noise, resulting in a denoised version of the surface that is visually compliant with the true structure of the surface (as

observed visually). For the 3DEMD denoising, we used a 2-ring neighborhood for the median filtering, $\lambda_M = 1 \times 10^6$, and $\theta = 0.5$.

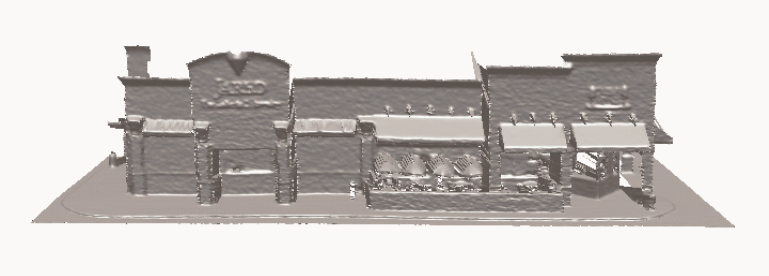
In contrast, the bottom two rows of Figure 8.1 shows the results of standard Laplacian smoothing and non-adaptive curvature based smoothing on the same data. Note that the Laplacian smoothed version of the data visibly distorts the surface model, destroying edge and detail information in its attempt to remove the surface noise. The non-adaptive curvature based algorithm preserves more of the surface information than the Laplacian smoothing, but still results in visible degradation of the detail areas and edges. Figure 8.2 shows a zoomed in region of the surface, giving a closer look at the noisy surface and the results of the 3 different methods' attempts at noise removal.

The next stage of the detail-enhancing denoising process is to use the geometry enhancement method from Chapter 5 to bring out the details embedded in the 3D signal. The process is the same as that in Chapter 5: regions of interest are identified, a model of the underlying data is constructed, and then the 3D geometry is enhanced in the areas of interest.

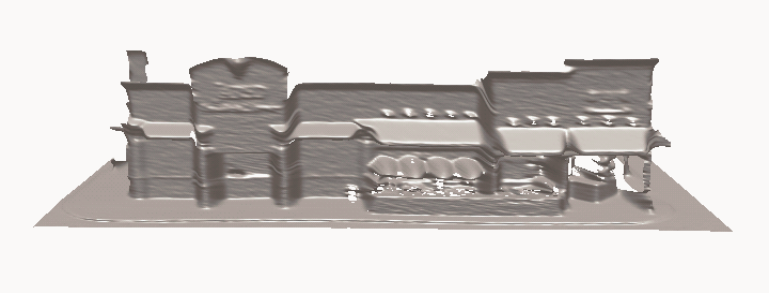
Figure 8.3 shows the results of the 3DEMD denoising with details refined around the "Jared" logo. The ROI that was processed is shown in blue on the top row of Figure 8.3. The middle row shows the denoised model with the details enhanced in the area defined by the ROI. Compare the results with the original surface and the smoothed surfaces of Figure 8.1., shown for comparison on the bottom row of Figure 8.3. The edges around the detail regions of the ROI are crisper than those of any from the denoised or smoothed versions.



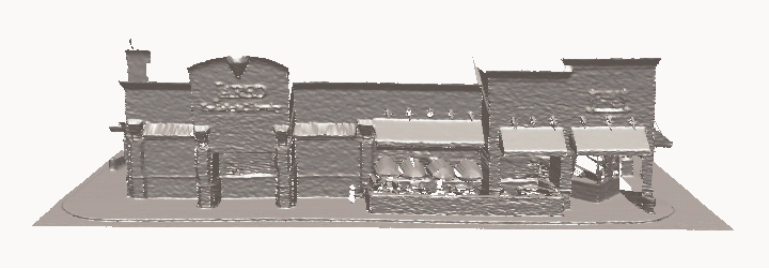
Original



3DEMD

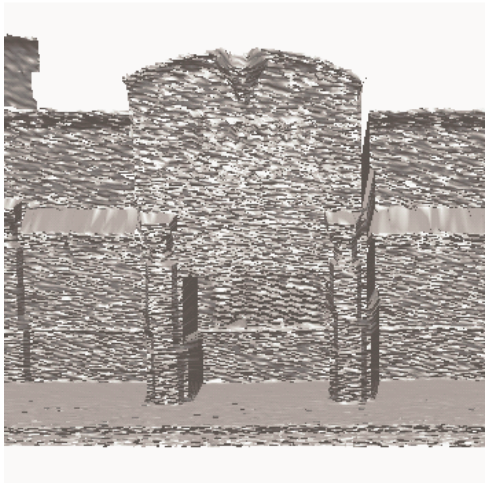


Laplacian

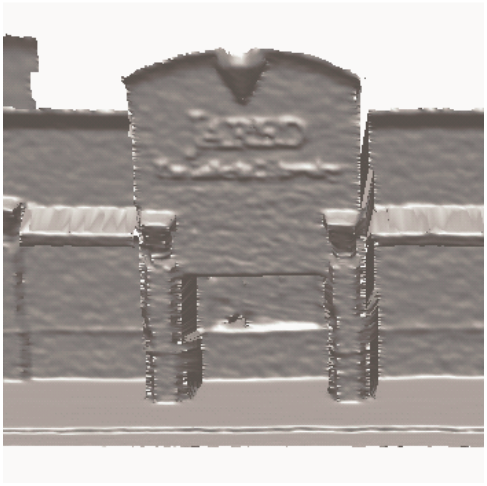


Non-adaptive Curvature

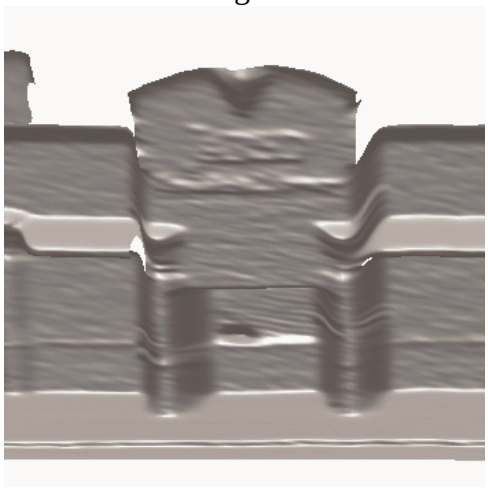
Figure 8.1. Comparison of the results from 3DEMD based smoothing of real mobile scanning data acquired from the UTK Mobile Scanning System with Laplacian smoothing and non-adaptive curvature based smoothing.



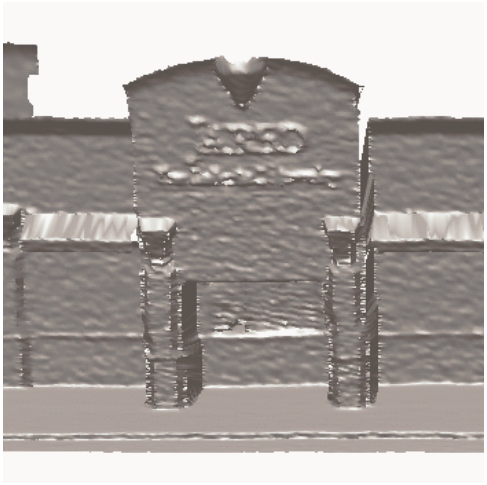
Original



3DEMD

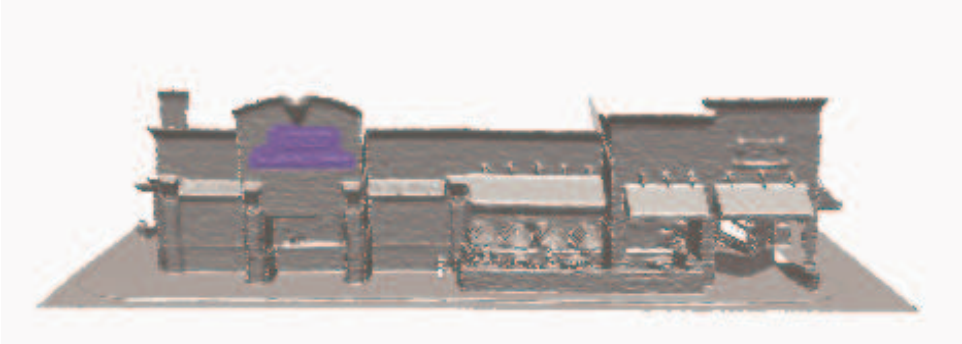


Laplacian

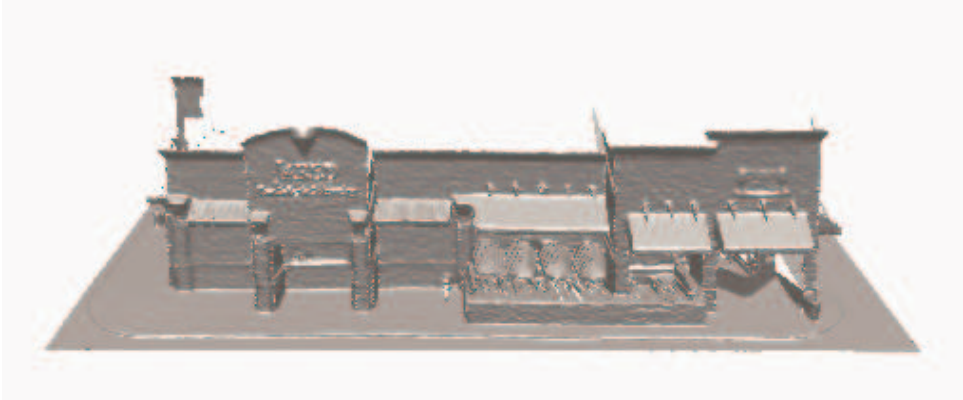


Non-adaptive

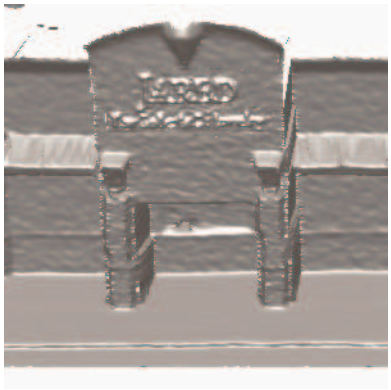
Figure 8.2. Zoomed in region of the data shown in Figure 8.1 for comparison purposes.



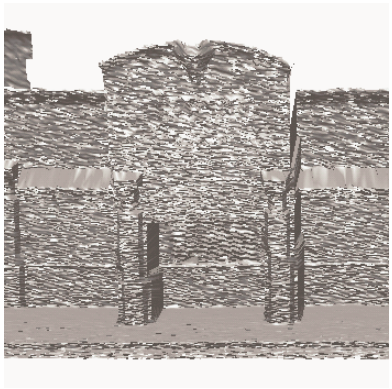
ROI



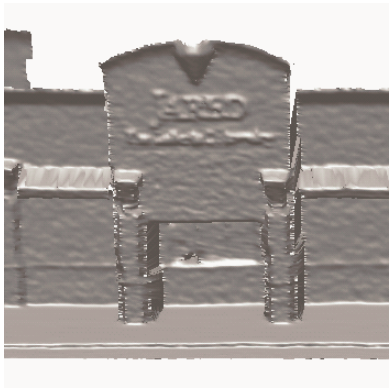
Detail Enhanced



Zoom of detail region



Zoom of Original Scan



Zoom of 3DEMD Denoised

Figure 8.3. Results of detail-enhancement and denoising of the StripMall dataset.

A quantitative comparison of our 3DEMD-based was performed using the mobile scanning simulator mentioned in Section 5.3 and a set of ground truth data. The 3D surface of the ground truth model was "scanned" using the simulator under a variety of noise effects. The resulting "raw" data was then denoised using our 3DEMD denoising and the denoised results compared with Laplacian smoothing and non-adaptive curvature smoothing of the same data.

As the ground truth for this example, we have used the Grand Canyon model shown in Figure 5.8. The model was used as the input to the mobile scanning simulator, with the scanning parameters chosen to simulate a scanning resolution of 360 m x 600 m (as in the experiments shown in Figures 5.8 and 5.9, as well as Table 5.2). The first experiment (shown in Figure 8.3) only adds Gaussian noise in the scanning direction (z-axis), with a 0 mean and a standard deviation of 10 (the approximate vertical resolution, typical error values for a laser scanner). The second experiment (shown in Figure 8.5) uses the same scanner noise parameters and adds a 5m GPS error to the scanner's location and a 2% vibration for the orientation.

Figure 8.3 shows the noisy (scanner only) output from the simulator and the results from the noise reduction algorithms. The adaptive denoising based on the 3DEMD analysis isolates and removes the scanner noise, while preserving the edge and detail information. The resulting surface is more visually true to the original than that produced by the Laplacian or non-adaptive curvature based solutions. Note again that the Laplacian smoothed version visibly distorts the surface by over smoothing in an attempt to get rid of the noise component. The reduced noise models have then been enhanced using our automatic detail enhancement algorithm in order to show the full effects of the noise reduction efforts.

The quantitative analysis portion of this experiment was performed by finding the closest point on the reconstructed surfaces for every point in the original model. The deviation from the reference model was recorded for every point and the tabulated results are given in Table 8.1. Notice that our 3DEMD-based method outperforms both the Laplacian and non-adaptive curvature smoothing methods. The deviations from the ground truth for each of the noise reduction models are plotted in Figure 8.4. For reference, the noisy scanned signal's deviations have also been provided. The errors for the 3DEMD-denoised and reconstructed signal are smaller than those from the other methods. These results show that the 3DEMD denoising and detail enhancement process can indeed yield 3D surfaces that are closer to the original models than is readily available from a scanned version of the data.

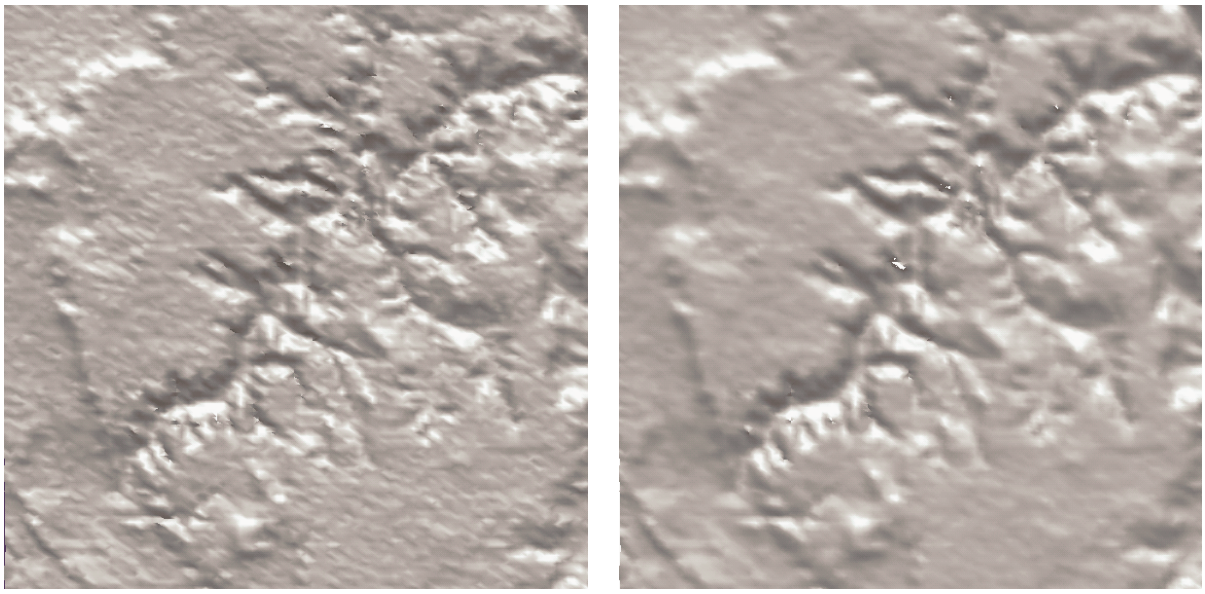
The results from the second experiment are shown in Figure 8.5. Here, position and orientation error has been added to the scanner noise from the previous experiment. As mentioned previously, a 5m standard deviation noise was added to the position of the sensor during the scanning process, and a 2% uncertainty in the pose estimation. The noise was removed in the same fashion as before, and the resulting surfaces are shown. Again, the adaptive denoising based on the 3DEMD analysis shows a better response at removing and isolating the system noise. When the reduced noise surface models are enhanced using our detail enhancement algorithm, the 3DEMD-based results are visually (and analytically) closer to the original model than either of the competing methods.

Quantitative error analysis was performed on the recovered surfaces in the same manner as in the previous experiment and the results are shown in Table 8.1. The 3DEMD algorithm once again outperformed Laplacian and non-adaptive curvature noise reduction methods to produce a reconstructed surface that more closely resembles the ground truth model. The color coded deviations from the ground truth model are shown in Figure 8.6. The errors for the 3DEMD-denoised and reconstructed signal are smaller than those from the other methods.

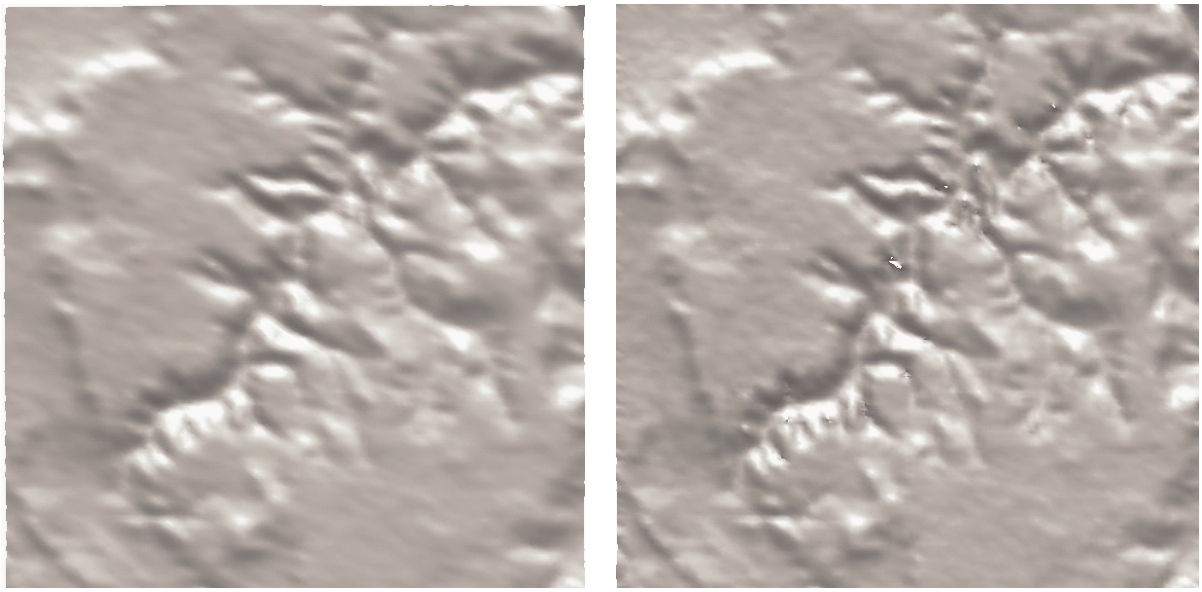
Given that the 3DEMD-based denoising has proven to result in reconstructed surfaces that contain less noise (error) than the other methods, there has been some question as to whether it is better to denoise first and then perform the detail enhancing, or vice versa. We set up an experiment regarding this question with the scan plus pose noise dataset. We used the enhanced noisy model seen in Figure 8.5 as the input to the 3DEMD denoising method and compared the output with the previous results. Figure 8.7 shows the output of the enhancement followed by denoising, side by side with the output from the denoising followed by enhancement. The results that denoising followed by enhancement yields the better results. The deviations from the original model for this experiment are also shown in Figure 8.7, and the tabulated results are shown in the last line of Table 8.1.

These results hold for this particular dataset and may not be valid for all possible data. However, it should be noted that while the computation complexity for the enhancement process for this dataset remained the same between the two experiments, the 3DEMD denoising required much more processing - approximately 100 times as many computations - to yield the final result, due to the need to perform the 3DEMD analysis on a 1024x1024 dataset, rather than the original 150x100 set. This leads to the observation that in most cases it will likely be a better idea to perform the denoising and then follow up with the enhancement process. Also, applying the denoising process first will, in many cases, help the ROI segmentation algorithm for the enhancement process to reduce the number and size of regions to be enhances, thus reducing the amount of computation time.

Furthermore, the baseline kriging process we use for enhancement is particularly sensitive to noise in the measurement process. The regularization term we added to accommodate this eases the requirements and allows the system to handle more local variation in the sampling process. However, this easement is in the sampling function value only. The sensitivity to the location of measurement remains. Applying the 3DEMD denoising before the enhancement process helps to alleviate the sample location errors before they propagate into the kriging system.

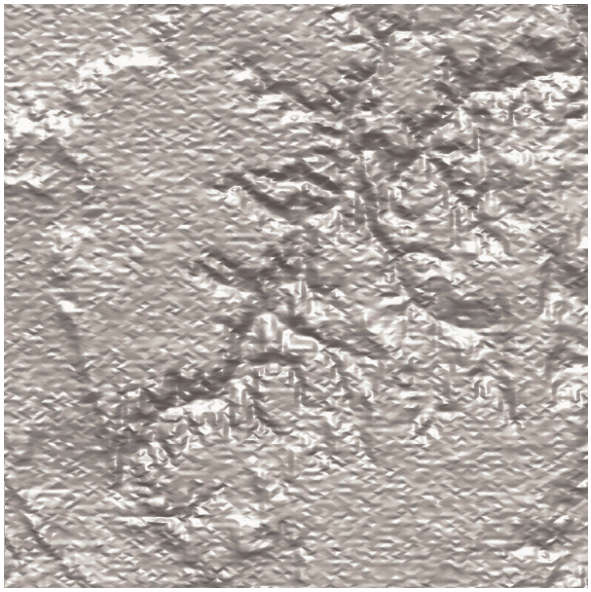


Sampled Noisy 3DEMD

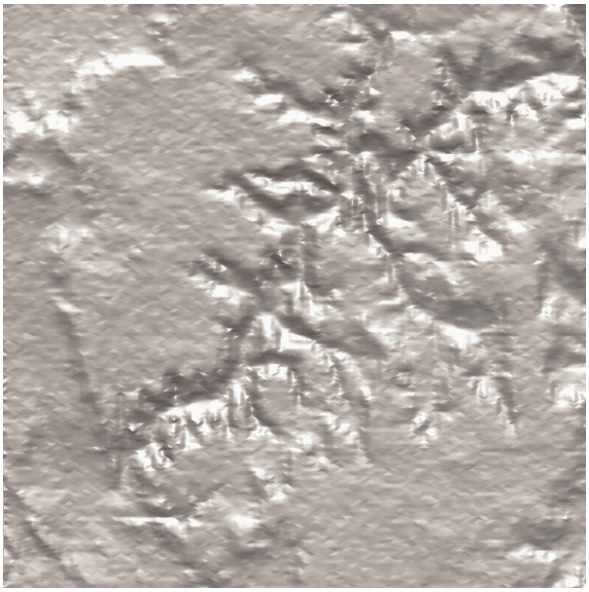


Laplacian Non-adaptive

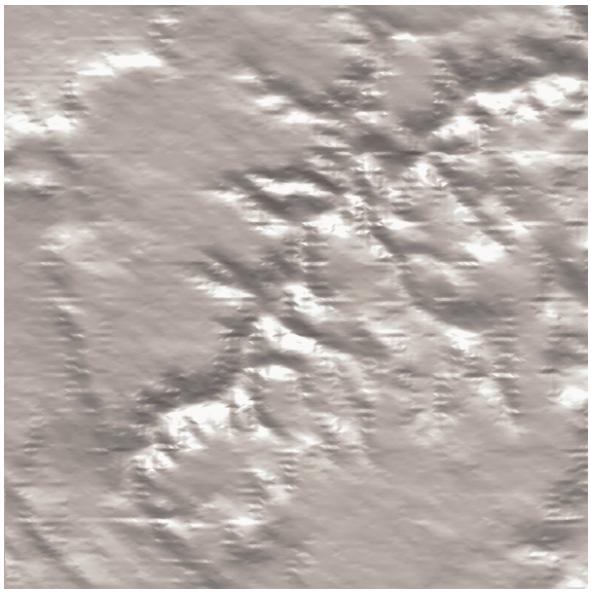
Figure 8.3. Comparison of 3DEMD-based denoising with Laplacian and non-adaptive curvature smoothing for a synthetic dataset consisting of a model with scanner noise added.



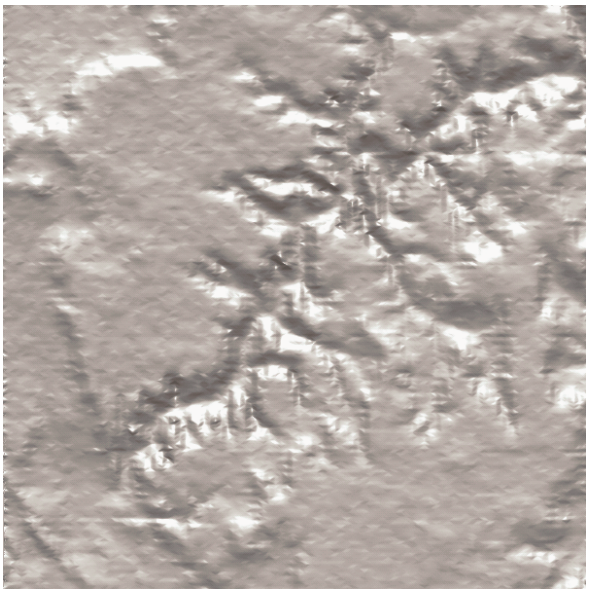
Enhanced Noisy



Enhanced 3DEMD



Enhanced Laplacian



Enhanced Non-adaptive

Figure 8.3. Continued.

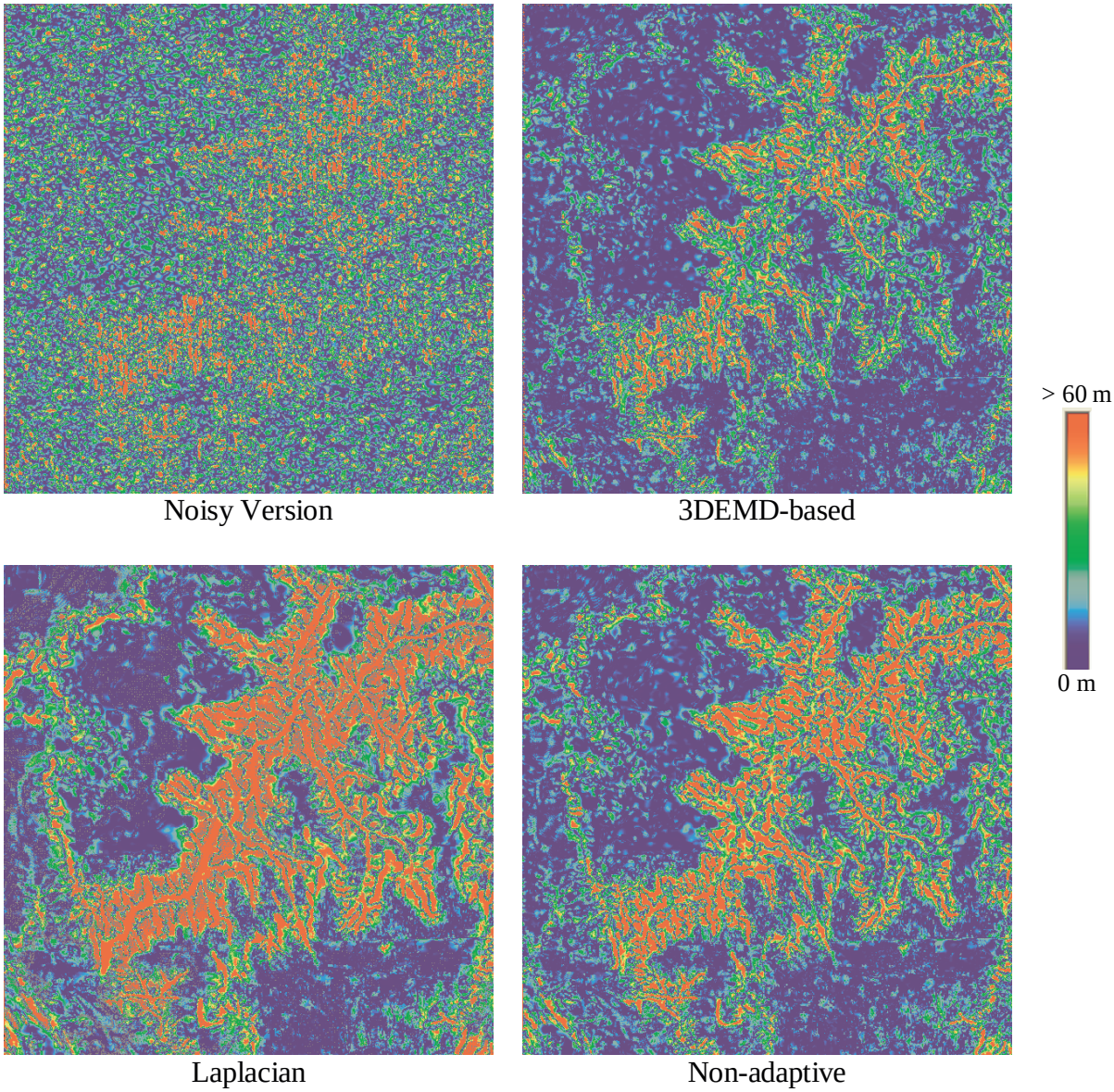
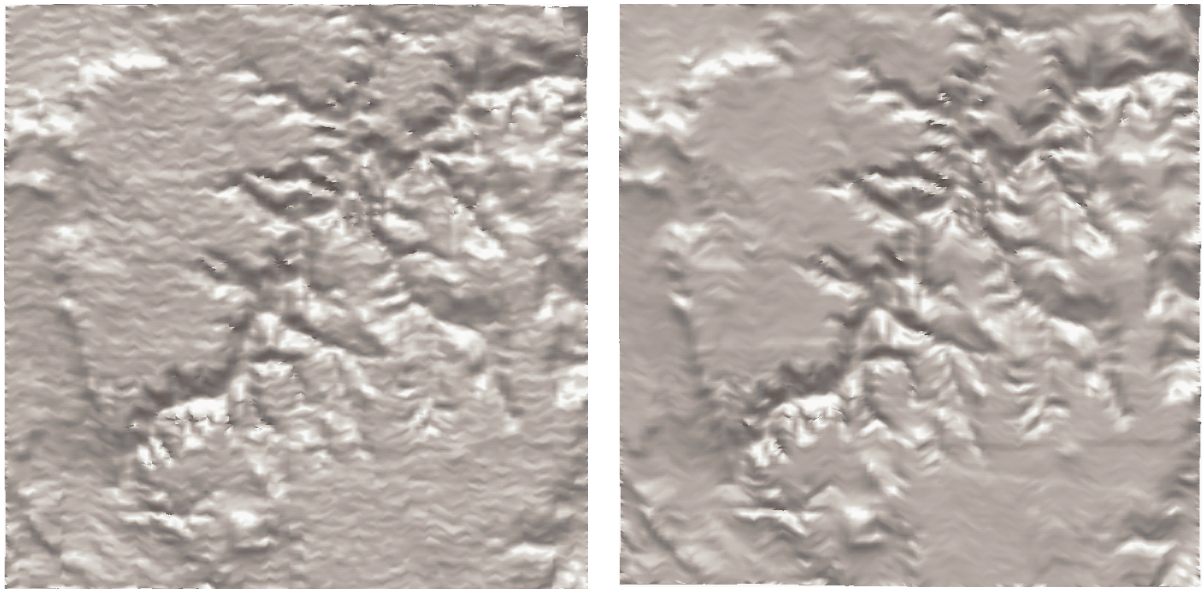
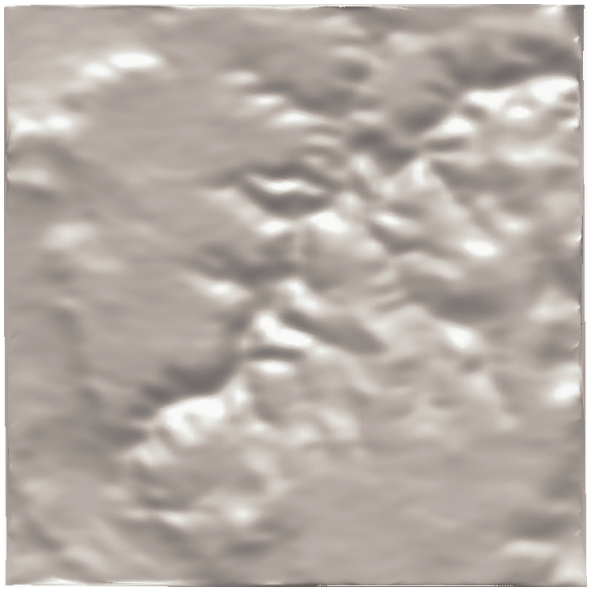


Figure 8.4. Color mapped deviation of denoised and enhanced models (Figure 8.3) from the ground truth dataset.

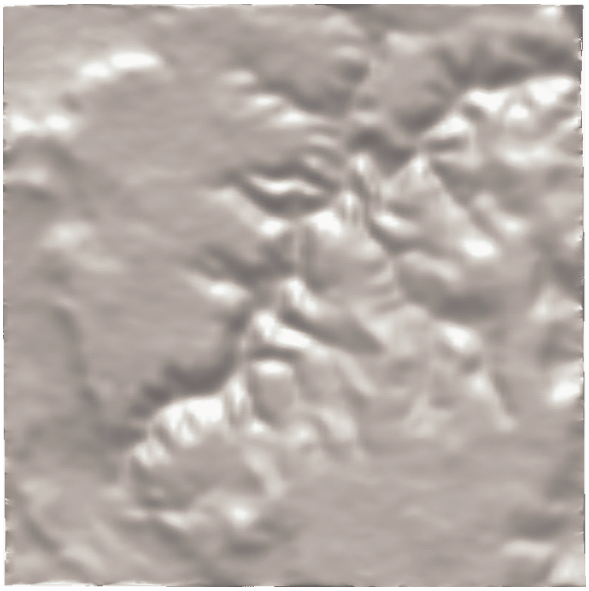


Pose + Scan Noise

3DEMD

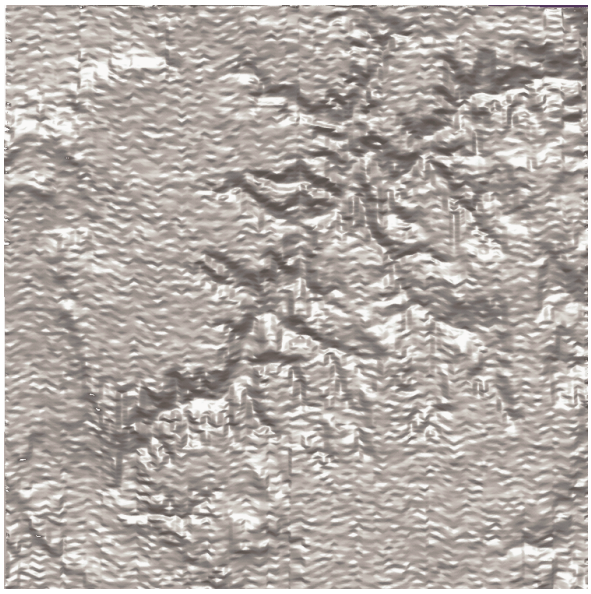


Laplacian

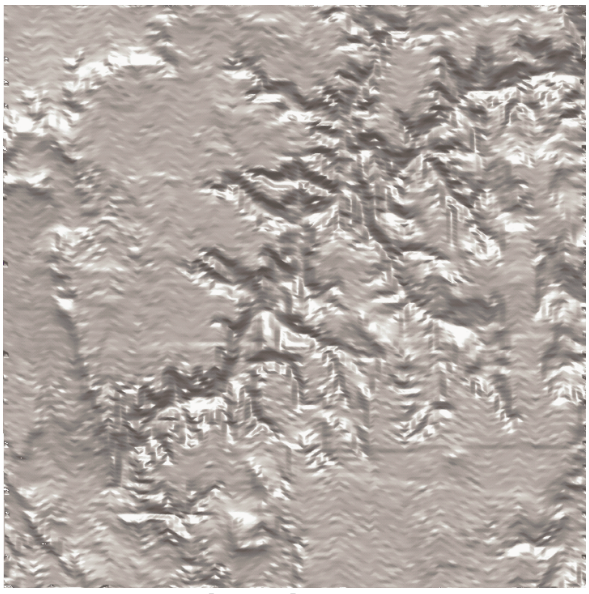


Non-adaptive

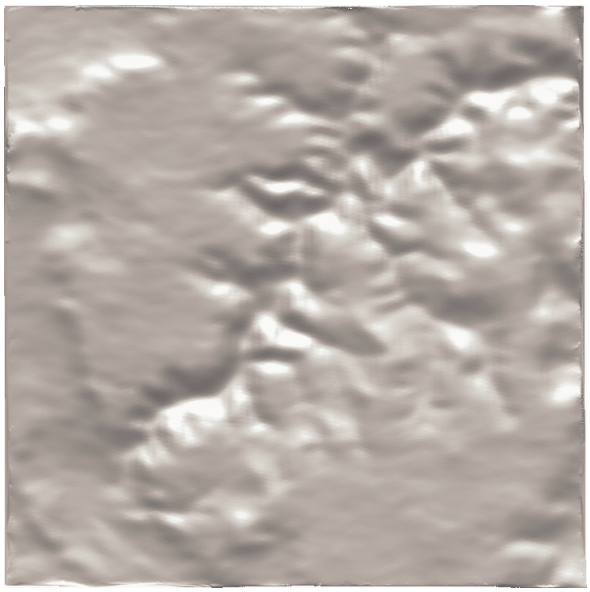
Figure 8.5. Comparison of 3DEMD-based denoising with Laplacian and non-adaptive curvature smoothing for a synthetic dataset consisting of a model with scanner, positioning, and orientation noise added.



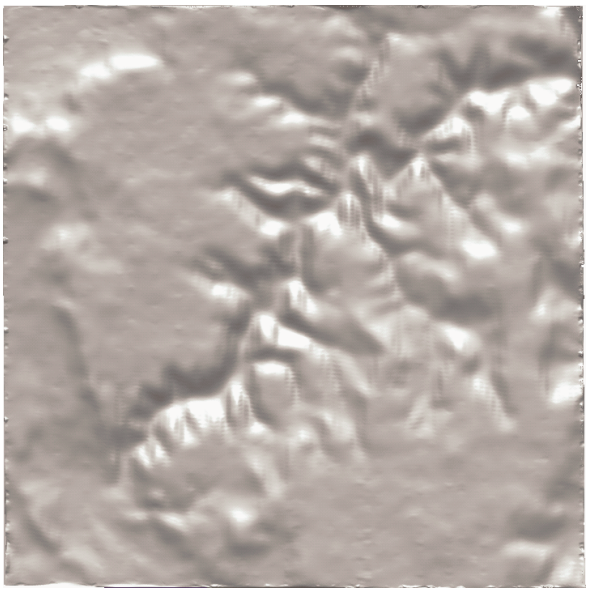
Enhanced Noisy



Enhanced 3DEMD



Enhanced Laplacian



Enhanced Non-adaptive

Figure 8.5. Continued.

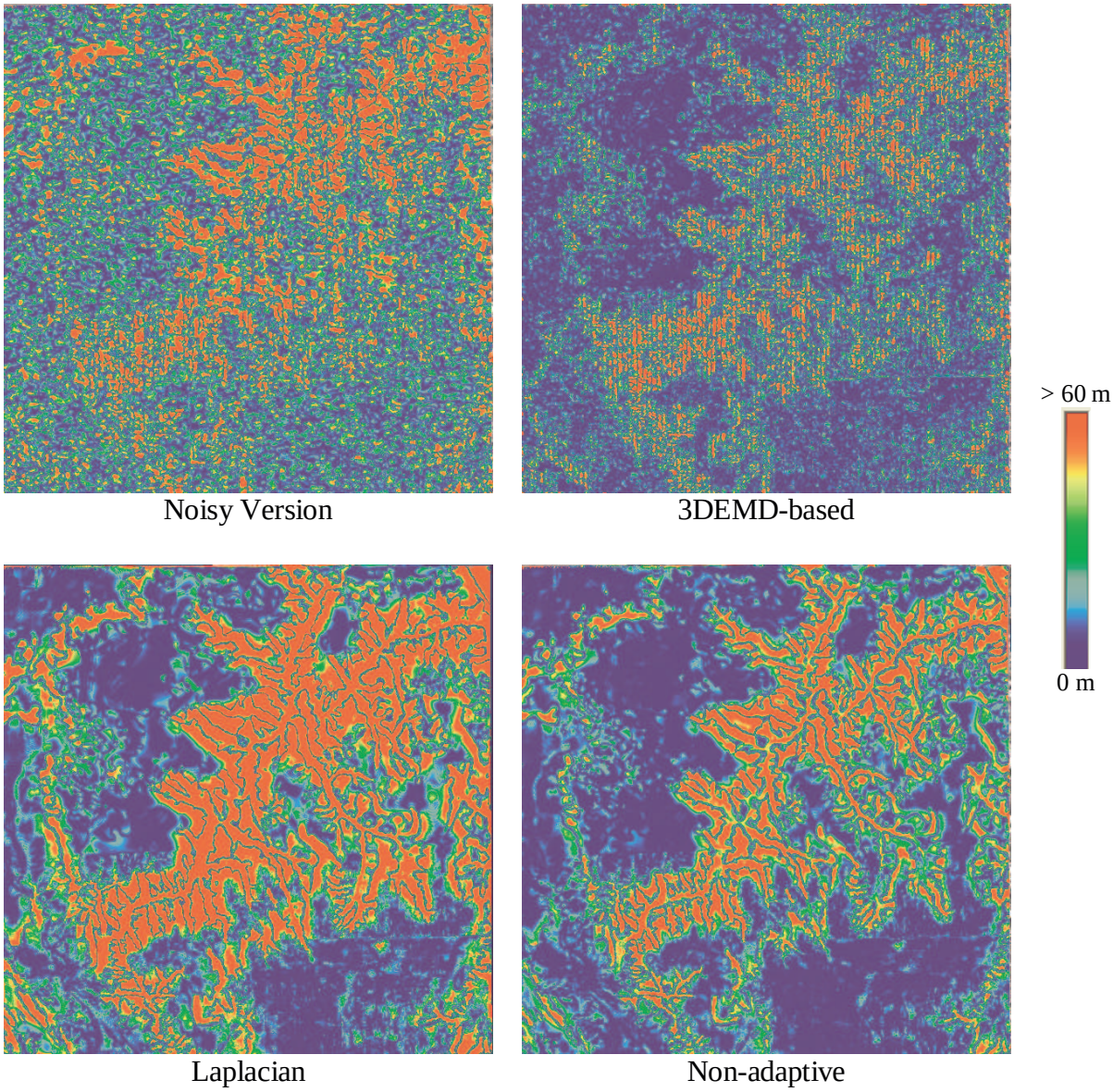


Figure 8.6. Color mapped deviation of denoised and enhanced models (Figure 8.5) from the ground truth dataset.

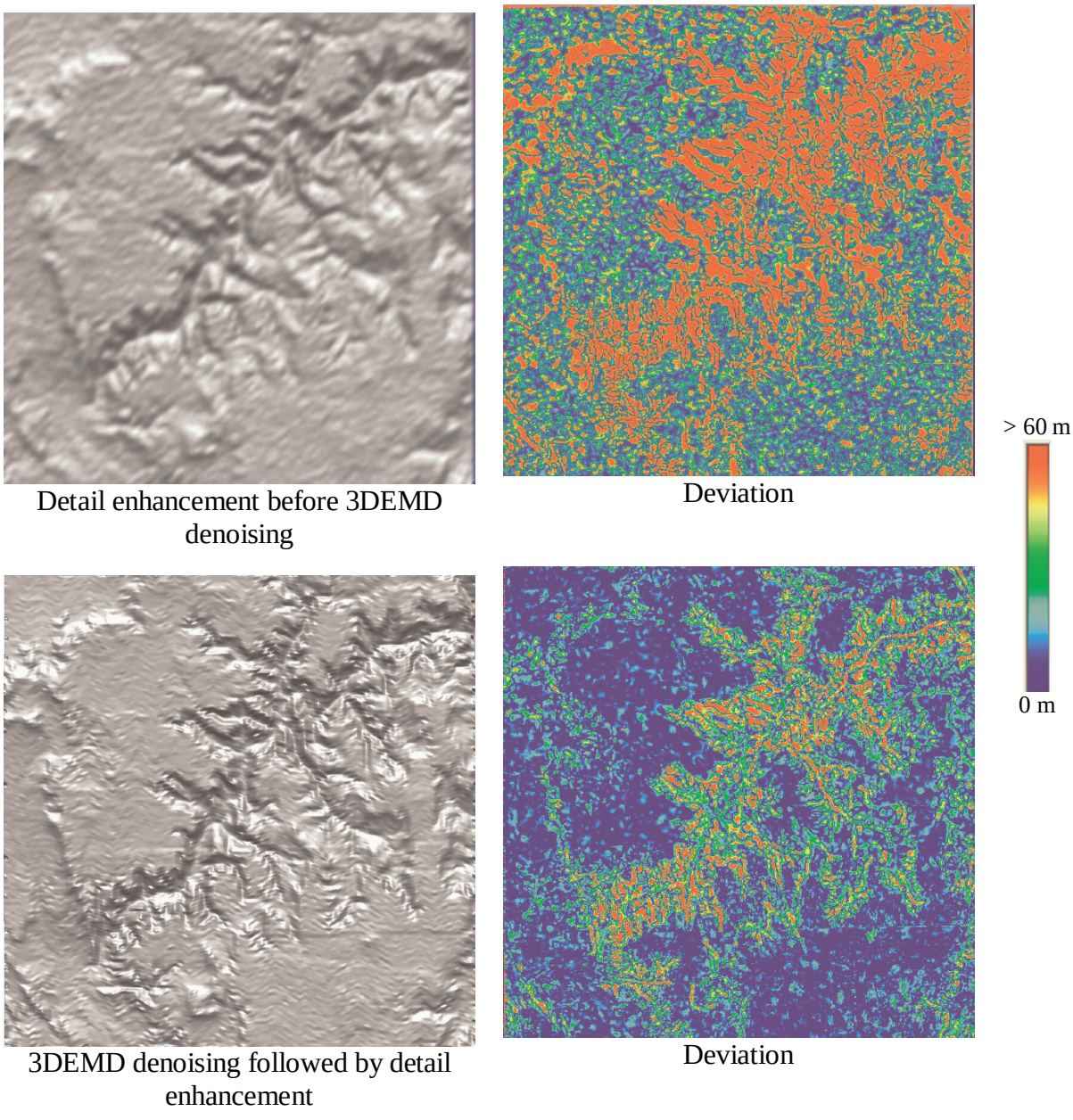


Figure 8.7. Comparison of detail enhancing denoising operation order. The top row contains a noisy surface from Figure 8.5 that was enhanced before applying the 3DEMD denoising, along with its deviation from the ground truth model. For comparison, the bottom row contains the surface resulting from applying the denoising before the detail enhancement.

Table 8.1. Quantitative comparison of 3DEMD-based denoising and detail enhancement with standard methods. Statistics shown are the Minimum, Maximum, Mean, and Median errors along with the Standard Deviation (SD), Coefficient of Variation (CV), and RMS of the errors.

Model Method	Min (mm)	Max (mm)	Mean (mm)	Median (mm)	SD	CV	RMS
Scanner noise only	0	446.6619	35.8898	20.4789	40.7475	1.1350	54.2982
Laplacian	0	493.3213	49.3862	28.1456	56.8786	1.1517	75.3273
Non-adaptive	0	441.3683	28.4944	21.9413	26.7246	0.9378	39.0658
3DEMD	0	428.8889	23.6960	16.6991	25.2113	1.0639	34.5997
Scanner and pose noise	0	127.8685	39.1964	32.2104	30.6614	0.7823	49.7664
Laplacian	0	432.7249	63.0574	34.0002	73.0267	1.1158	96.4839
Non-adaptive	0	128.4273	52.0914	29.5457	61.0460	1.1719	80.8304
3DEMD	0	135.8978	31.1176	22.1647	29.0418	0.9315	42.6078
Enhancement followed by 3DEMD	0	394.2934	42.3020	27.9963	35.6139	0.8419	53.8163

9 Conclusion & Future Work

The overall goal of this work was to develop a general detail-enhancing denoising algorithm for 3D surfaces based on leveraging a proven technique for locally refining the acquired surface geometry with a novel algorithm for 3D surface decomposition to identify the components of the 3D model that are most affected by the digitization uncertainty. The vehicle for demonstrating this work is the Mobile Scanning System developed here at the IRIS Laboratory. Significant contributions that came out of this work are the 3D surface decomposition method used to identify the "surface frequencies" for use in the denoising process, and the automation process for locally refining surface regions containing small-scale details, including the use of regularization to account for the uncertainty involved in measuring the "known" data points. In addition, we also have a contribution in self-localization from video sequences, in the form of a novel false match rejection algorithm, as presented in Chapter 4.

9.1 Dissertation Key Points

The key points for the foundation of this research are:

A New False Match Detection Algorithm to aid Pose Estimation From Video

False matches play an important role in the estimation of pose from video. In fact, they can be the killer for an otherwise perfect pose estimation algorithm, as the output of any algorithm is only as good as its input. We developed an outlier rejection method based on the commonality of track direction of the true matches. Our outlier rejection method decreased the false matches passed to the pose estimator by 50% or more, as compared to the standard method of false match rejection. Furthermore, the rejection of outliers is linear in order, once an initial setup stage of finding the track direction is complete.

Automated Detail Enhancement for Mobile Scanning Data though Modified Kriging

With any mobile 3D scanning solution, there is a tradeoff between acquisition time and acquired data resolution. Ideally, the experimental setup would allow you to capture data at

the highest resolution required by the application, with no additional costs involved. In reality, the costs involved can be significant. Our solution to managing the tradeoff between acquisition time and acquired data density is a software-based surface resolution enhancement algorithm that builds a statistical model of the data that was acquired, followed by a ehancement algorithm that optimally inserts new data estimates to populate the 3D surface to the desired resolution. The method is a refinement of standard kriging, modified to provide automation for surface segmentation, model determination, and surface estimation. The algorithm was demonstrated on a variety of data types, showing its efficacy for detail enhancement in a variety of application settings.

A Extension of Empirical Mode Decomposition to 3D Surfaces

The most important contribution of this research is perhaps the development of an entirely new 3D signal analysis tool based on Empirical Mode Decomposition. The extension of the 1D signal analysis technique to attributes recorded on a 3D model surface yields a powerful tool for 3D modeling and analysis applications. The 3DEMD tool provides the following benefits:

1. 3DEMD fills a gap in the 3D surface modeling and analysis toolbox, providing an encapsulated method for analyzing attribute functions attached to a 3D surface.

2. Measured and constructed attributes attached to a 3D surface can be readily analyzed in a frequency band setting similar to that of a Fourier decomposition. The frequency bands are extracted from the data itself, rather than an arbitrary function. This allows the researcher to determine attribute importance across a broad spectrum of frequencies.

3. 3DEMD is an intuitive extension of a well-known and widely applied 1D signal analysis technique. It's precursors have found solid niches of application in the last half decade, lending evidence that this 3D extension will do likewise.

4. The implicit surfacing approach to representing the surface attributes on the 3D mesh is ideal for further processing tasks, including: denoising, band selection, data fusion, and feature detection.

Signal Source Separation and Detail-enhancing Denoising of 3D Surfaces

Perhaps the most telling aspect of a new tool's functionality is the uses to which it can be put. While tools that have only a single use can have value, the general tool that can be used in a variety of situations becomes the most valued over time. In an effort to present the potential benefits of 3DEMD as a useful, and general, tool for 3D signal analysis, we demonstrated its performance as applied to two mainstream signal processing areas: source separation and denoising. The results for source separation showed that 3DEMD indeed has the potential to

play a major role in signal decomposition efforts. Those results led to the detail enhancing denoising process, where the final high frequency band noise was identified through source separation by 3DEMD analysis and the noise reduction processing was performed only on that component. The denoised version showed the noise impact removed or greatly reduced, while the regions of high detail (logos, hard edges, etc.) were enhanced and protected from the smoothing process.

9.2 Unanswered Questions and Avenues of Future Efforts

No research project is ever considered to be truly complete by its progenitors. There are always "what if"s and "what about"s that naturally occur at every stage of research. Therefore we do not believe we have addressed every issue involved with this research in its entirety. There are several avenues that could further be explored towards improving the final results, especially in the optimization and surface generation aspects. For example, we chose to use radial basis functions to develop our implicit function surfaces for the 3DEMD sifting process. However, there are dozens of surfacing algorithms that may provide equal or better results for this task. We showed the efficacy of the base implementation and leave it to future research efforts to improve upon our basis.

This work opens up several promising opportunities for further research. Of particular interest is the identification of additional application areas that can and will benefit from the 3DEMD extension. Our efforts at using this powerful analysis tool provided a step in the right direction, and will hopefully open up further application opportunities in the future. In addition, the application of 3DEMD to point clouds points towards a broader venue of 3D signal analysis in the areas of biomedical imaging and computed tomography, where volume sets are used instead of tessellated meshes.

For outlier rejection, the use of the common behavior of the feature tracks for identifying true matches proved useful in the orthogonal viewing angle case. Extending this algorithm to investigate the curl and divergence of the feature tracks could provide false match rejection results for the non-orthogonal viewing angle case. Specifically, many mobile robot systems use cameras aligned in the axis of motion for navigation purposes. The curl/divergence solution could potentially provide a large benefit in the video localization algorithms for these systems.

References

A. Adam, E. Rivlin, and I. Shimshoni, "ROR: Rejection of outliers by rotations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23**(01), pp. 78-84, 2001.
N. Amenta, M. Bern, and M. Kamvysselis, "A new voronoi-based surface reconstruction algorithm," Proceedings of the *SIGGRAPH*, pp. 415 - 421, 1998.
M. E. Antone and S. Teller, "Automatic recovery of relative camera positions in urban scenes," *Computer Vision and Pattern Recognition*, **2**, pp. 282 - 289, 2000.
X. Armangue, H. Araujo, and J. Salvi, "A review on egomotion by means of differential epipolar geometry applied to the movement of a mobile robot," *Pattern Recognition*, **36**, pp. 2927-2944, 2003.
C. Baillard and A. Zisserman, "Automatic reconstruction of piecewise planar models from multiple views," Proceedings of the *Proc. Computer Vision and Pattern Recognition*, pp. 559 - 565, 1999.
C. L. Bajaj, F. Bernardini, and G. Xu, "Automatic reconstruction of surfaces and scalar fields from 3D scans," Proceedings of the *SIGGRAPH*, pp. 109 - 118, 1995.
S. T. Barnard and M. A. Fischler, "Computational Stereo," *ACM Computing Surveys*, **14**(04), pp. 553-572, 1982.
J. Bauer, A. Klaus, K. Karner, C. Zach, and K. Schindler, "MetropoGIS: A feature based city modeling system," *Photogrammetric Computer Vision*, **B**, pp. 22 - 27, 2002.
F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *IEEE Transactions on Visualization and Computer Graphics*, **5**, 1999.
P. Besl and N. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14**, pp. 239 - 256, 1992.
F. Bignone, R. Weiss, and E. Riseman, "Automatic extraction of generic house roofs from high resolution aerial imagery," Proceedings of the *Computer Vision - ECCV'96*, vol. 1, pp. 85-96, 1996.
J. Blinn, "Simulation of wrinkled surfaces," *Computer Graphics*, pp. 192-198, 1978.
J. D. Bossler and R. W. Schmidley, "Airborne integrated mapping system (AIMS): Recent results in applications for large-scale mapping," Proceedings of the *Urban and Regional Information Systems Association*, 1997.
A. Branca, E. Stella, and A. Distante, "Autonomous navigation of underwater vehicles," Proceedings of the *International Conference on Intelligent Robots and Systems*, vol. 2, pp. 1113-1118, 1998.
A. Branca, E. Stella, and A. Distante, "Passive navigation using egomotion estimates," *Image and Vision Computing*, **18**, pp. 833-841, 2000.
C. Brenner and N. Haala, "Fast production of virtual reality city models," Proceedings of the *IAPRS: GIS - Between Visions and Applications*, vol. 32, pp. 77-84, 1998.
E. A. Bretz, "X marks the spot, maybe," *IEEE Spectrum*, **37**(04), pp. 26-36, 2000.
J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans, "Reconstruction and representation of 3D objects with radial basis functions," Proceedings of the *SIGGRAPH 2001*, pp. 67-76, 2001.
H. R. Cho, K. M. Lee, and S. U. Lee, "A new robust 3D motion estimation under persective projection," Proceedings of the *International Conference on Image Processing*, 2001.
N. R. Chrisman, "The role of quality information," Proceedings of the *AUTOCART06*, vol. 2, pp. 303-321, 1983.

N. R. Chrisman, "The error component in spatial data," *Geographical Information Systems*, **1**, pp. 165-174, 1991.
S. G. Chroust and M. Vincze, "Fusion of vision and inertial data for motion and structure estimation," *Journal of Robotic Systems*, **21**(02), pp. 73-83, 2004.
C.-S. Chua, Y. K. Ho, and Y. Liang, "Rejection of mismatched correspondences along the affine epipolar line," *Image and Vision Computing*, **18**, pp. 445-462, 2000.
R. L. Cook, "Shade trees," *Computer Graphics*, pp. 223-231, 1984.
S. Coorg and S. Teller, "Extracting textured vertical facades from controlled close-range imagery," *Computer Vision and Pattern Recognition*, **1**, pp. 625 - 632, 1999.
P. Corke, "An inertial and visual sensing system for a small helicopter," *Journal of Robotic Systems*, **21**(02), pp. 43-51, 2004.
K. T. Coughlin and K. K. Tung, "11-Year solar cycle in the stratosphere extracted by the empirical mode decomposition method," *Advances in Space Research*, **34**(02), pp. 323-329, 2004.
N. Cressie, *Statistics for Spatial Data*. New York, NY: Wiley-Interscience, 1991.
N. A. C. Cressie, "Fitting variogram models by weighted least squares," *Mathematical Geology*, **17**, pp. 563-286, 1985.
M. J. E. Crombaghs, R. Brüglemann, and E. J. deMin, "On the adjustment of overlapping strips of laseraltimeter height data," *Photogrammetry and Remote Sensing*, **33**(B3/1), pp. 230-237, 2000.
Y. Cui and S. S. Ge, "Autonomous vehicle positioning with GPS in urban canyon environments," *IEEE Transactions on Robotics and Automation*, **19**, pp. 15-25, 2003.
E. Dalcin, "Data quality concepts applied to taxonomic databases," in *Biological Sciences*: University of Southampton, 2004.
J. C. Davis, *Statistics and data analysis in geology*, 2nd ed. New York: Wiley, 1986.
M. Desbrun, M. Meyer, P. Schroeder, and A. H. Barr, "Implicit fairing of irregular meshes using diffusion and curvature flow," *Computer Graphics*, **33**, pp. 317 - 324, 1999.
H. Q. Dinh, G. Turk, and G. Slabaugh, "Reconstructing surfaces by volumetric regularization using radial basis functions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(10), pp. 1358-1371, 2002.
N. Dyn, D. Levin, and J. A. Gregory, "A Butterfly subdivision scheme for surface interpolation with tension control," *ACM Transactions on Graphics*, **9**, pp. 160-169, 1990.
S. F. El-Hakim, P. Boulanger, F. Blais, J. A. Beraldin, and G. Roth, "A mobile system for indoors 3-D mapping and positioning," Proceedings of the *Optical 3-D Measurement Techniques*, pp. 275 - 282, 1997.
O. Faugeras, *Three-Dimensional Computer Vision*. Cambridge Massachusetts: The MIT Press, 1993.
O. Faugeras, L. Robert, S. Laveau, G. Csurka, C. Zeller, C. Gauclin, and I. Zoghlami, "3-D reconstruction of urban scenes from image sequences," *Computer Vision and Image Understanding*, **69**(03), pp. 292 - 309, 1998.
P. Flandrin, G. Rilling, and P. Concalves, "Empirical Mode Decomposition as a filter bank," *Signal Processing Letters*, **9**(06), pp. 181-184, 2004.
D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*. Upper Saddle River, NJ: Prentice Hall, 2003.

R. Franke and G. M. Nielson, "Smooth interpolation of large sets of scattered data," *International Journal for Numerical Methods in Engineering*, **15**(11), pp. 1691-1704, 1980.
D. Frere, J. Vandekerckhove, T. Moons, and L. VanGool, "Automatic modeling and 3D reconstruction of urban buildings from aerial imagery," Proceedings of the *IEEE International Geoscience and Remote Sensing Symposium*, pp. 2593 - 2596, 1998.
P. Fricker, R. Sandau, and A. S. Walker, "Digital aerial sensors - A new approach," Proceedings of the *ISPRS Workshop on Sensors and Mapping from Space*, 1999.
C. Früh and A. Zakhor, "3D model generation for cities using aerial photographs and ground level laser scans," *Computer Vision and Pattern Recognition*, **2**, pp. 31 - 38, 2001a.
C. Früh and A. Zakhor, "Fast 3D model generation in urban environments," Proceedings of the *International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 165 - 170, 2001b.
C. Früh and A. Zakhor, "Data processing algorithms for generating textured 3D building façade meshes from laser scans and camera images," Proceedings of the *Intl. Symposium on 3D Data Processing, Visualization, and Transmission*, pp. 834 - 847, 2002.
A. Fusiello, E. Trucco, T. Tommasini, and V. Roberto, "Improving feature tracking with robust statistics," *Pattern Analysis & Applications*, **2**, pp. 312-320, 1999.
A. Georgiev and P. K. Allen, "Localization methods for a mobile robot in urban environments," *IEEE Transactions on Robotics*, **20**(05), pp. 851-864, 2004.
T. Gotoh, M. Kudo, J. Toyama, and M. Shimbo, "Geometry reconstruction of urban scenes by tracking vertical edges," Proceedings of the *International Conference on Knowledge Based Intelligent Information Engineering Systems*, pp. 455 - 458, 1999.
S. Graovac, "Principles of fusion of inertial navigation and dynamic vision," *Journal of Robotic Systems*, **21**(01), pp. 13-22, 2004.
M. S. Grewal, L. R. Weill, and A. P. Andrews, *Global Positioning Systems, Inertial Navigation, and Integration*. New York: John Wiley & Sons, 2001.
A. Gruen and X. Wang, "Urban 3-D mapping for a hybrid GIS," Proceedings of the *Urban 3D/Multi-Media Mapping*, 1999.
I. Guskov, W. Sweldens, and P. Schroeder, "Multiresolution signal processing for meshes," Proceedings of the *SIGGRAPH*, pp. 325-334, 1999.
H.Hoppe, "Smooth view-dependent level-of-detail control and its application to terrain rendering," *IEEE Visualization*, pp. 35-42, October 1998.
D. Hähnel, W. Burgard, and S. Thrun, "Learning compact 3D models of indoor and outdoor environments with a mobile robot," *Robotics and Autonomous Systems*, **44**, pp. 15-27, 2003.
P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems*. Philadelphia, PA: SIAM, 1998.
C. Harris and M. Stephens, "A combined corner and edge detector," Proceedings of the *Alvey Vision Conference*, pp. 147-151, 1988.
R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*: Cambridge University Press, 2000.
R. I. Hartley, "In defense of the 8-point algorithm," Proceedings of the *International Conference on Computer Vision*, pp. 1064-1070, 1995.

H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," Proceedings of the *SIGGRAPH*, pp. 71 - 78, 1992.
H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Piecewise smooth surface reconstruction," Proceedings of the *SIGGRAPH*, pp. 295 - 302, 1994.
N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and the Hilbert spectrum for non-linear and non-stationary time series analysis," *Proceedings of the Royal Society of London A.*, **454**, pp. 903-995, 1998.
I. INUS Technologies, *Rapidform demonstration models*, http://www.rapidform.com, 2007.
C. Jaynes, "View alignment of aerial and terrestrial imagery in urban environments," Proceedings of the *International Workshop on Integrated Spatial Databases*, vol. 1737, pp. 3-19, 1999.
T. Jebara, A. Azarbayenjani, and A. Pentland, "3D structure from 2D motion," *Signal Processing Magazine*, **16**, pp. 66 - 84, 1999.
A. E. Johnson, C. Yang, and L. H. Matthies, "Machine vision for autonomous small body navigation," Proceedings of the *IEEE Aerospace Conference*, vol. 7, pp. 661 - 671, 2000.
O. Jokinen, "Building 3-D city models from multiple unregistered profile maps," Proceedings of the *Recent Advances in 3-D Digital Imaging and Modeling* pp. 242 - 249, 1997.
S. J. Julier and H. F. Durrant-White, "On the role of process models in autonomous land vehicle navigation systems," *IEEE Transactions on Robotics and Automation*, **19**, pp. 1-14, 2003.
V. Kamath, Y.-C. Lai, L. Zhu, and S. Urval, "Empirical mode decomposition and blind source separation methods for antijamming with GPS signals," Proceedings of the *IEEE Position Location and Navigation Symposium*, pp. 335-341, 2006.
S. Karbacher and G. Häusler, "A new approach for modeling and smoothing of scattered 3D data," Proceedings of the *SPIE: Three-Dimensional Image Capture and Applications*, vol. 3313, pp. 168-177, 1998.
S. Karbacher, S. Seeger, and G. Häusler, "Refining triangle meshes by non-linear subdivision," Proceedings of the *3-D Digital Imaging and Modeling*, pp. 270-277, 2001.
H. Ki, J. Shin, J. Jung, S. Lee, and J. Paik, "Surface smoothing for enhancement of 3D data using curvature-based adaptive regularization," in *Combinatorial Image Analysis*, vol. 3322/2004, *Lecture Notes in Computer Science*: Springer Berlin/Heidelberg, 2004, pp. 488-501.
L. Kobbelt, "A variational appraoch to subdivision," *Computer Aided Geometrical Design*, **13**, pp. 743-761, 1996.
L. Kobbelt and P. Schröder, "A multiresolution framework for variational subdivision," *ACM Transactions on Graphics*, **17**(04), pp. 209-237, 1998.
S. C. Lee, S. K. Jung, and R. Nevatia, "Integrating ground and aerial views for urban site modeling," *International Conference on Pattern Recognition*, **IV**, pp. 107 - 112, 2002.
J. J. Leonard and H. F. Durrant-Whyte, "Simultaneous map building and localization for an autonomous robot," Proceedings of the *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, vol. 3, pp. 1442-1447, 1991.
A. Linderhed, "2D empirical mode decompositions in the spirit of image compression," Proceedings of the *SPIE: Wavelet and Independent Component Analysis Applications IX*, vol. 4738, pp. 1-8, 2002.

Y. Liu, X. Zhang, and T. S. Huang, "Estimation of 3D structure and motion from image corners," *Pattern Recognition*, **36**, pp. 1269-1277, 2003.
J. Lobo, L. Almeida, J. ALves, and J. Dias, "Registration and segmentation for 3D map building: A solution based on stereo vision and inertial sensors," Proceedings of the *International Conference on Robotics and Automation*, 2003.
C. Loop, "Smooth spline surfaces over irregular meshes," Proceedings of the *Computer Graphics*, pp. 303-310, 1994.
W. Lorenson and H. Cline, "Marching cubes: A high resolution 3D surface reconstruction algorithm," Proceedings of the *SIGGRAPH*, pp. 163 - 169, 1987.
M. Lounsbery, T. DeRose, and J. Warren, "Multiresolution analysis for surfaces of arbitrary topological type," *ACM Transactions on Graphics*, **16**(01), pp. 34-73, 1997.
B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," Proceedings of the *International Joint Conference on Artificial Intelligence*, pp. 674-679, 1981.
H.-G. Maas, "Fast determination of parametric house models from dense airborn laser scanner data," *International Archives of Photogrammetry and Remote Sensing*, **32**(2W/1), pp. 1-6, 1999.
J. I. Maletic and A. Marcus, "Data cleansing: Beyond integrity analysis," Proceedings of the *Conference on Information Quality*, pp. 200-209, 2000.
S. Mallat, *A Wavelet Tour of Signal Processing*, 2nd ed: Academic Press, 1999.
D. Manandhar and R. Shibasaki, "Geo-referencing of multi-sensor range data for vehicle-borne laser mapping system (VLMS)," 2002.
A. T. Martins, P. M. Q. Aguiar, and M. A. T. Figueiredo, "Navigating in Manhattan: 3D orientation from video without correspondences," Proceedings of the *International Conference on Image Processing*, 2003.
F. Masson, J. Guivant, and E. Nebot, "Robust navigation and mapping architecture for large environments," *Journal of Robotic Systems*, **20**(10), pp. 621-634, 2003.
T. Moons, D. Frere, J. Vandekerckhove, and L. V. Gool, "Semi-automatic generation of 3D city models for urban sites from aerial imagery," Proceedings of the *AGI Conference at GIS 98*, pp. 37-41, 1998.
L.-P. Morency and R. Gupta, "Robust real-time egomotion from stereo images," Proceedings of the *International Conference on Image Processing*, 2003.
E. P. S. Neto, J. C. Cejka, M. A. Custaud, P. Abry, J. Frutoso, C. Gharib, and P. Flandrin, "Assessment of cardiovascular autonomic control by the Empirical Mode Decomposition," *Methods of Information in Medicine*, **43**, pp. 60-65, 2004.
NOAA, *National Oceanic and Atmospheric Administration - Home Page* http://www.noaa.gov, 2004.
M. Notomi, S. Ozawa, and H. Zen, "Modeling of urban scene by motion analysis," *Systems and Computers in Japan*, **31**(02), pp. 1 - 9, 2000.
J. C. Nunes, Y. Bouaoune, E. Delechelle, O. Niang, and P. Bunel, "Image analysis by bidimensional empirical mode decomposition," *Image and Vision Computing*, **21**, pp. 1019-1026, 2003.
J. Nygårds, P. Skoglar, M. Ulvklo, and T. Högström, "Navigation aided image processing in UAV surveillance: Preliminary results and design of an airborne experimental system," *Journal of Robotic Systems*, **21**(02), pp. 63-72, 2004.

P. J. Oonincx and J.-P. Hermand, "Empirical mode decomposition of ocean acoustic data with constraint on the frequency range," Proceedings of the *European Conference on Underwater Acoustics*, pp. 739-744, 2004.

D. L. Page, A. F. Koschan, and M. A. Abidi, "Perception-based 3D triangle mesh segmentation using fast marching watersheds," Proceedings of the *International Conference on Computer Vision and Pattern Recognition*, vol. II, pp. 27-32, 2003.

J. Peng, V. Strela, and D. Zorin, "A simple algorithm for surface denoising," Proceedings of the *IEEE Visualization*, pp. 107-112, 2001.

M. Pollefeys, L. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, and J. Tops, "Video-to-3D," Proceedings of the *Symposium on Photogrammetric Computer Vision*, vol. A, pp. 252 - 257, 2002.

M. Pollefeys, R. Koch, and L. V. Gool, "Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters," *International Journal of Computer Vision*, **32**, pp. 7 - 25, 1999.

M. Pollefeys, R. Koch, M. Vergauwen, and L. V. Gool, "Automated reconstruction of 3D scenes from sequences of images," *ISPRS Journal of Photogrammetry & Remote Sensing*, **55**, pp. 251 - 267, 2000.

Riegl-USA, *Riegl USA-Scanners & Rangefinders for Surveying, Altimetry and other applications*, 2002.

G. Rilling, P. Flandrin, and P. Goncalves, "On empirical mode decomposition and its algorithms," Proceedings of the *IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, 2003.

E. Rivlin, I. Shimshoni, and E. Smolyar, "Image-based robot navigation in unknown indoor environments," Proceedings of the *International Conference on Intelligent Robots and Systems*, vol. 3, pp. 2736-2742, 2003.

M. Roy, S. Foufou, A. Koschan, F. Truchutet, and M. Abidi, "Multiresolution analysis for irregular meshes," Proceedings of the *SPIE: Photonics East*, vol. 5266, pp. 249-259, 2003.

S. Roy and V. Govindu, "MRF solutions for probabilistic optical flow formulations," Proceedings of the *International Conference on Pattern Recognition*, vol. 3, pp. 1041-1047, 2000.

J. Salvi, X. Armangue, and J. Pages, "A survey addressing the Fundamental Matrix estimation problem," Proceedings of the *International Conference on Image Processing*, pp. 209-212, 2001.

C. Schmid, R. Mohr, and C. Bauckhage, "Comparing and evaluating interest points," Proceedings of the *International Conference on Computer Vision*, pp. 230-235, 1998.

C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors," *International Journal of Computer Vision*, **37**, pp. 151 - 172, 2000.

V. Sequeira, K. Ng, E. Wolfart, J. G. M. Goncalves, and D. Hogg, "Automated reconstruction of 3D models from real environments," *ISPRS Journal of Photogrammetry and Remote Sensing*, **54**, pp. 1 - 22, 1999.

A. Shamir, L. Shapira, and D. Cohen-Or, "Mesh analysis using geodesic mean-shift," *The Visual Computer*, **22**(02), pp. 99-108, 2006.

L. Shapiro, A. Zisserman, and M. Brady, "Motion from point matches using affine epipolar geometry," Proceedings of the *European Conference on Computer Vision*, pp. 73-84, 1994.

D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," Proceedings of the *23rd ACM National Conference*, pp. 517-524, 1968.

J. Shi and C. Tomasi, "Good features to track," Proceedings of the *International Conference on Computer Vision and Pattern Recognition*, pp. 593-600, 1994.

I. Shylakhter, M. Rozenoer, J. Dorsey, and S. Teller, "Reconstructing 3D tree models from instrumented photographs," in *IEEE Computer Graphics and Applications*, vol. 21, 2001, pp. 53-61.

B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. London, UK: Chapman & Hall, 1986.

D. A. Simon, M. Hebert, and T. Kanade, "Real-time 3-D pose estimation using a high-speed range sensor," Proceedings of the *International Conference on Robotics and Automation*, vol. 3, pp. 2235-2241, 1994.

I. Stamos and P. Allen, "3-D model construction using range and image data," Proceedings of the *Computer Vision and Pattern Recognition*, pp. 531 - 536, 2000.

B. M. Straub and C. Heipke, "Automatic extraction of trees for 3D-city models from images and height data," in *Automatic Extraction of Man-Made Objects from Aerial and Space Images (III)*: Biirkhauser, 2001.

D. M. Strong, Y. W. Lee, and R. W. Wang, "Data quality in context," *Communications of ACM*, **40**, pp. 103-110, 1997.

Y. Sun, D. L. Page, J. K. Paik, A. Koschan, and M. A. Abidi, "Triangle mesh-based edge detection and its application to surface segmentation and adaptive surface smoothing," Proceedings of the *International Conference on Image Processing*, vol. 3, pp. 825-828, 2002.

H. Surmann, A. Nüchter, and J. Hertzberg, "An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments," *Robotics and Autonomous Systems*, **45**, pp. 181-198, 2003.

I. Suveg and G. Vosselman, "Automatic 3D building reconstruction," Proceedings of the *SPIE: Photonics West 2002 - Electronic Imaging*, vol. 4661, pp. 59 - 69, 2002.

G. Taubin, "Estimating the tensor of curvature of a surface from a polyhedral approximation," Proceedings of the *International Conference on Computer Vision*, pp. 902 - 907, 1995a.

G. Taubin, "A signal processing approach to fair surface design," Proceedings of the *SIGGRAPH*, 1995b.

S. Teller, M. Antone, Z. Bodnar, M. Bosse, S. Coorg, M. Jethwa, and N. Master, "Calibrated, registered images of an extended urban area," Proceedings of the *Computer Vision and Pattern Recognition*, vol. 1, pp. 813 - 820, 2001.

S. M. Thayer, M. Dias, B. Nabbe, B. L. Digney, M. Hebert, and A. Stentz, "Distributed robotic mapping of extreme environments," Proceedings of the *SPIE: Mobile Robots XV and Telemanipulator and Telepresence Technologies VII*, vol. 4195, pp. 84 - 95, 2001.

T. Tommasini, A. Fusiello, E. Trucco, and V. Roberto, "Making good features to track better," Proceedings of the *International Conference on Computer Vision and Pattern Recognition*, pp. 178-183, 1998.

E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision*. Upper Saddle River, NJ: Prentice-Hall, Inc., 1998.
S. Tzeng, H.-C. Huang, and N. A. Cressie, "A fast, optimal spatial-prediction method for massive datasets," *Journal of the American Statistical Association*, **100**, pp. 1343-1357, 2005.
M. Uehara and H. Zen, "From digital map to 3D map: Creating 3D map by motion stereo utilizing 2D map," Proceedings of the *IAPR Workshop on Machine Vision Applications*, pp. 592 - 595, 2000.
K. Usher, P. Ridley, and P. Corke, "Visual servoing of a car-like vehicle - An application of omnidirectional vision," Proceedings of the *International Conference on Robotics and Automation*, pp. 4288-4293, 2003.
R. Vidal, Y. Ma, S. Hsu, and S. Sastry, "Optimal motion estimation from multiview normalized epipolar constraint," Proceedings of the *International Conference on Computer Vision*, 2001.
G. Vosselman and S. Dijkman, "3D building model reconstruction from point clouds and ground plans," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, **34**(3/W4), pp. 37-44, 2001.
G. Wahba, "Spline models for observational data," Proceedings of the *CBMS-NSF Regional Conference Series in Applied Mathematics*, pp. 11-14, 45-65, 1990.
M. P. Wand and M. C. Jones, *Kernel Smoothing*. London, UK: Chapman & Hall, 1995.
X. Wang and A. R. Hanson, "Extracting surface textures and microstructures from multiple aerial images," Proceedings of the *Computer Vision and Pattern Recognition*, pp. 301 - 306, 1997.
Y. M. Wang, B. S. Peterson, and L. H. Staib, "3D brain surface matching based on geodesics and local geometry," *Computer Vision and Image Understanding*, **89**, pp. 252-271, 2003.
X. Zhang and D. Zhao, "Range image segmentation via edges and critical points," *Proc. SPIE:*, **2501**, pp. 1626 - 1637, 1995.
Z. Zhang, "Determining the epipolar geometry and its uncertainty: A review," *International Journal of Computer Vision*, **27**(02), pp. 161-198, 1998.
Z. Zhang and Y. Shan, "Incremental motion estimation through modified bundle adjustment," Proceedings of the *International Conference on Image Processing*, 2003.
H. Zhao and R. Shibasaki, "High accurate positioning and mapping in urban area using laser range scanner," Proceedings of the *IEEE Intelligent Vehicles Symposium*, 2001a.
H. Zhao and R. Shibasaki, "Reconstructing urban 3D model using vehicle-borne laser range scanners," Proceedings of the *International Conference on 3D Digital Imaging and Modeling*, pp. 349 - 356, 2001b.
H. Zhao and R. Shibasaki, "A robust method for registering ground-based laser range images of urban outdoor objects," *PE&RS*, **67**(10), pp. 1143-1153, 2001c.
D. Zimmerman, C. Pavlik, A. Ruggles, and M. P. Armstrong, "An experimental comparison of ordinary and universal kriging and inverse distance weighting," *Mathematical Geology*, **31**(04), pp. 375-390, 1999.
S. Zinger, M. Nikolova, M. Roux, and H. Maître, "3D Resampling for Airborne Laser Data of Urban Areas," Proceedings of the *Photogrammetric Computer Vision*, vol. A, pp. 418-423, 2002.

A. Zisserman, A. W. Fitzgibbon, and G. Cross, "VHS to VRML: 3D graphical models from video sequences," Proceedings of the *International Conference on Multimedia and Systems*, pp. 51 - 57, 1999.
D. Zorin, P. Schröder, and W. Sweldens, "Interpolating subdivision for meshes with arbitrary topology," Proceedings of the *SIGGRAPH'96*, pp. 189-192, 1996.

Vita

Brad Grinstead was born in Lakenheath, England on September 20, 1976. He attended Texas Tech University in Lubbock, TX, where he received a dual Bachelor's degree in both Electrical Engineering and Computer Science in 1999. In May of 2001 he graduated from Texas Tech University with a Master of Science degree in Electrical Engineering, with a field specialization of image processing. He joined the IRIS Lab at The University of Tennessee in the summer of 2001, where he enrolled in the Ph.D. program in the Department of Electrical Engineering. Since then, he has worked in the IRIS lab as a Graduate Research Assistant, focusing on mobile 3D scanning and processing solutions for computer vision tasks. Near the end of that term, he took a position as a Senior Image Scientist at IAVO Research and Scientific, where he is a research team leader and technical advisor for a number of image analysis and knowledge mining research projects. He earned his Doctor of Philosophy degree in Electrical Engineering in December 2007.